



## **WORK PACKAGE (WP) 2**

### **“WEB DEVELOPMENT”**



Project “Youth networking for economic exchange in the cross-border region”  
Project No. CB007.2.22.078

---

The project is co-funded by EU through the Interreg-IPA CBC Bulgaria–Serbia  
Programme

This publication has been produced with the assistance of the European Union through the Interreg-IPA CBC Bulgaria-Serbia Programme, CCI No 2014TC16I5CB007. The contents of this publication are the sole responsibility of National Association Legal Initiative for Open Government and can in no way be taken to reflect the views of the European Union or the Managing Authority of the Programme.



## CONTENTS:

L. WEB DEVELOPMENT BASICS, PRINCIPLES, APPROACHES .....	3
2. FRONT-END WEB DEVELOPMENT: HTML, CSS, JAVASCRIPT, JQUERY, BOOTSTRAP .....	9
WHAT YOU ALREADY NEED TO KNOW .....	18
3. BACK-END WEB DEVELOPMENT: JAVA, MYSQL .....	21
4. AJAX ANGULAR.JS PROGRAMMING .....	25
5. WEBSITE DEVELOPMENT AND MODIFICATION VIA CMS.....	31
6. DEVELOPMENT ENVIRONMENTS: INTEGRATED DEBUGGING ENVIRONMENT (IDE), DEVELOPMENT TOOLS, TOOLS TO SUPPORT METHODOLOGICAL CONTROL OF SOURCES, ERROR TRACKING, AUTOMATED TOOLS FOR IMPLEMENTATION/CONTINUOUS INTEGRATION, ETC. ....	35
7. DATABASES: DEVELOPMENT OF BASIC SKILLS OF STUDENTS IN ANALYZING AND APPLYING APPROPRIATE MODELS FOR DATA PROCESSING .....	38
8. INTERFACE DESIGN: DEVELOPMENT OF STUDENTS' COMPETENCIES FOR ENTERING COMPLEX SCENARIOS FOR USE AND INDEPENDENT PARTICIPATION IN THE DESIGN PROCESS RELATED TO THE DESIGN OF COMPLEX USER INTERFACES.....	40
PRACTICAL EXERCISES AND TESTS .....	49
CSS.....	56
JAVASCRIPT.....	62
JQUERY .....	63
BOOTSTRAP .....	66
CSS.....	90
JAVASCRIPT.....	96
JQUERY .....	97
BOOTSTRAP .....	100



## 1. Web development basics, principles, approaches

Despite its great popularity, the term “user experience” (“UX”) remains poorly understood by many website owners. In short, the user experience is the interaction of Internet users with a website, regardless of the purpose of their visit - to find specific information, to buy something, etc. Often the user experience is measured by the ability of a user to achieve their goal efficiently and easily, without errors and obstacles. Web designers and programmers should use techniques that focus on the user experience and influence the visual and technical part of the website. Before you start your project, you need to research the users who will use your website - who they are, why they are your users, how they interact with the product, etc. You need to find out which methods work and which do not - for this purpose, you can ask users what they would like to see changed. This will help you get an idea of their behavior.

In order to fully understand what a user experience design (UX) is, let me tell you what UX designers do - something that can be done by describing the five main components of UX:

- Information architecture (IA)
- Interactive design
- Usability
- Prototyping
- Visual design

### 1. Information architecture

Information architecture is the way to make a connection between people and content in the most understandable way for them. A hierarchy is created in the content of the page and various elements are used, such as navigation to create a structure. Different types of information from the context are taken and organized in a way that makes it easier to find content.

### 2. Interactive design

The interactive design addresses the specific interactions between users and the screen. The graphic design serves the purposes of users, implemented by interactive design to represent the brand, using graphics, images, fonts, colors, icons, etc.

The interactive design also uses prototyping to define the specific behavior and functions of the various components. For example: when designing mobile applications, should the login page hide, fade, or slide to the right? This type of transition must be explored in the interactive concept so that the final product can be as close as possible to the designer’s idea.

### 3. Usability



Usability uses data to determine the validity of design solutions. While any company has people of different backgrounds talking about business goals or specific engineering constraints, the role of the UX designer is to feel the user's feelings and understand the inconveniences they encounter while using the product.

This information can be obtained in a variety of ways - from focus groups to surveys, from laboratory usability studies to face-to-face interviews and site visits with eye tracking, map sorting, A/B testing, telemetry, etc.

Usability testing tools for quick and early feedback.

In this post, I'll explain how you can conduct remote or on-site usability testing sessions to evaluate your design and achieve the best you can.

Usability testing helps to understand the space of the product, while defining the behavior and needs of the user. At a high level is good practice to evaluate your hypotheses with real users and to measure and evaluate the changing product to confirm that it will meet the needs of users.

#### **4. Prototyping**

The prototype can be considered as a preliminary version, from which the following forms will be developed. It also offers a replication method that is based on feedback from stakeholders and users during usability testing.

This gives you an idea of the functionality of your design and what changes are needed to make your work enjoyable and functional. This is a way to set design priorities, test different options cheaply, and then understand logistical constraints and implementation conflicts. Prototyping is especially important because it brings you closer to the actual functionality of the product before you invest time, resources and money in development.

#### **5. Visual design**

Visual design involves the use of the product's vision to enhance the user experience. While visual design is not all that needs to be designed, it is still an important part of a well thought out product. It speaks volumes about the company's image and can influence the desire and commitment that the product generates.

The topic of visual design can be polarizing, as it is sometimes thought that if designers prefer aesthetics they will sacrifice usability. Great visual design can even make us ignore usability issues. New research shows that aesthetics and usability can be seen as positively related.

The visual design shows us how the product works through color, visual hierarchy, typography and thus is an important part of the overall user experience.

#### **Which are the elements of a successful UX design?**



To have a positive effect, your UX design must have several mandatory elements. You can find their brief presentation in the following lines.

### **Usability**

The usability of a website is directly related to its navigation and content. Generally speaking, navigation should be sufficiently clear and logical and lead to relevant content. For this purpose, you can use appropriate “call-action”, descriptive titles and submenus. The elements listed so far should be in reasonable number - neither too little nor too much.

If you do not find the golden mean, the user experience may not be good and may even lead to a refusal of the services and products you offer.

### **Accessibility**

The content, in addition to being informative and well organized, should be easily accessible to all users of the website. It itself, must meet several characteristics:

There should be no pages on the website that do not work.

The individual pages of the website should load quickly.

The website should be able to be well visualized on all the most commonly used browsers and devices (desktop and mobile).

### **Findability**

Your website should be easy for users to find. To achieve this, you need to use all the techniques of digital marketing - social networking, email marketing, content optimization (SEO optimization), etc. You should not be left with the impression that once you are successful with a part of your website, it is going to last forever. Search engines often change their algorithms, which require constant updating of the content.

### **Usefulness**

Too often, companies do not pay enough attention to the quality of the content they publish on their website. However, to ensure the best possible user experience, you need to pay special attention to it. Remember that the content must be unique, informative and different. This way, users will be satisfied with their stay on the website and will continue to return to it.

### **Attractiveness**

Another essential element of the user experience for a website is its attractiveness. It is directly related to the visual elements (photos, videos, typography, colors, color schemes, the arrangement of the elements) and the way they present the product.



## Reliability

Websites must show and prove to users that they are reliable. This can be achieved by providing them with relevant and credible content presented in the right tone.

**To own a website, you need three things: domain name, web hosting and developed website.**

**Domain** is the Internet address of the website - for example [www.vashetoime.com](http://www.vashetoime.com). It is paid for a minimum of one year and thanks to it users have access to your website. There are many domain areas, the most common and used are .com (commercial), .org (organization), .net (network), .eu (European Union), and .bg (Bulgaria), which, however, are significantly more expensive and require submitting additional documentation on your part.

A domain name is not something physical that you can touch or see. This is a series of characters that give your website identity (yes, name, as a person and business). Examples of domain name: [Google.com](http://Google.com), [Alexa.com](http://Alexa.com), [Linux.org](http://Linux.org), [eLearningEuropa.info](http://eLearningEuropa.info), and [Yahoo.co.uk](http://Yahoo.co.uk).

The Domain Name System (DNS) has a hierarchy of names. Top level domains (TLD) are set of common names in the hierarchy - COM, NET, ORG, EDU, INFO, BIZ, CO.UK, etc.

Example 1:

[Google.com](http://Google.com), [Linux.org](http://Linux.org), [Yahoo.co.uk](http://Yahoo.co.uk)

Note that these domains end up with a different “extension” (.com, .org, .co.uk.)? These extensions are known as TLD.

Official list of all domains of top level is kept by the Internet Assigned Numbers Authority (IANA) in the Root Zone Database. As of April 2018 there are a total of 1,532 TLD.

Some TLD are usually visible -

BIZ, BR, CA, CN, CO, CO.JP, COM.SG, COM.MY, EDU, ES, FR, INFO, MOBI, TECH, RU, United Kingdom, USA,

Some are less known

AF, AX, BAR, BUSINESS, BID, EXPERT, GURU, JOBS, MOBI, TECH, ESTATE, WIEN, WTF, WOW, XYZ

While most of these TLD are opened for public registration, there are strict recommendations for specific domain registration. For example, registration of a top-level domain (such as .co.uk for the United Kingdom) is restricted to nationals; and website activities regarding such domains are governed by local regulations and cyber laws.



Some extensions of these TLD are used to describe the “features” on the website - such as BIZ for business, EDU for education (schools, universities, colleagues, etc.), ORG for public organization, and top-level domain names with country code for locations. ICANN publishes case studies on the variety of generic TLD, check if this is interesting to you.

### **What is a top-level domain (ccTLD)?**

All domain names are unique. This means that there can only be one alexa.com in the world. You cannot register a name after it has been registered by others (managed by ICANN).

### **Rules for ccTLDs**

For those users trying to register a country-specific domain name option (such as “.us” or “.co.uk”), much of the registration process will be dedicated to whether the customer is a resident of that country and therefore legally allowed to purchase one of its country-specific top-level domains (will be discussed later). And that should hit a secondary point at home for users.

While there are hundreds of domain suffixes available (such as “.com” or “.net”), many of these domains have specific registration requirements.

For example, only organizations can register a “.org” domain name, and only US citizens can register a domain name ending with “.us”. Failure to comply with the instructions and requirements for each type of domain during the actual registration and payment process will result in the domain name being “released” back into the group of available domain names; the customer will have to choose the top-level domain for which he is actually responsible or cancel his purchase altogether.

During the registration process, it is also important to have information directly from the web host, as this information will be needed when filling out the DNS and MX records information during registration.

These two records determine the contents of the web hosting server when the user navigates to the domain, as well as how to address, send and receive email using this hosting package and the associated domain name. Inaccurate information will lead to errors and failed page loads.

### **Domain vs. subdomain**

Take mail.yahoo.com for example - yahoo.com is the domain, and in this case mail.yahoo.com is the subdomain.

The domain must be unique (for example, there may be only one Yahoo.com) and must be registered with a domain registrar (i.e. NameCheap and Path); while for subdomains, users are free to add it to the existing domain as long as their web host provides the service. Some





would say that subdomains are “third-level” domains in the sense that they are simply “sub-folders” in the domain’s root directory, typically used to organize the content of your website in different languages or different categories.

However, this is not the case in many cases, including in search engines - it is known that search engines (namely Google) treat the subdomain as a different domain, independent of the main domain.

**Hosting** is the physical storage of files on a website on a server that is permanently connected to the Internet, providing users with access to your site. More commonly, the term “web hosting” refers to a company that rents out its computer/servers to store your website and provides Internet connectivity so that other users can access the files on your website.

In most cases, these hosting companies will deal with server maintenance, such as backups, root configuration, maintenance, disaster recovery, etc.

### **How does web hosting work?**

Think of it as a house where you store all your things; but instead of storing your clothes and furniture, you store digital files (HTML, documents, images, videos, etc.) in a web host.

More commonly, the term “web hosting” refers to a company that rents out its computer/servers to store your website and provides Internet connectivity so that other users can access the files on your website.

Usually, a web hosting company does more than just storing your site. Here are some value-added services and features you can expect from your hosting provider:

Domain registration - so you can buy and manage a domain and hosting from the same provider

Website builder - Drag and drop the web editing tool to create a website

Email hosting - To send and receive emails from email@yourdomain.com

Basic hardware (server setup) and software (CMS, server OS, etc.) support

### **Web hosting vs. Data center**

The term “web hosting” usually refers to the server that hosts your website or the hosting company that rents the server space for you.

Data centers typically refer to the facility used to host servers.

The data center can be a room, a house or a very large building equipped with backup or backup power supplies, redundant data connections, environmental control - i.e. air conditioning, fire extinguishing and security devices.





## 2. Front-end Web Development: HTML, CSS, JavaScript, JQuery, Bootstrap

Web front-end is a term that describes the part of a web application or page that is visible to the end user. The aim of front-end developers is to create Internet sites and web applications based on Internet technologies such as HTML, CSS, JavaScript, Flash, etc., in the part that goes to the user's browser. They deal with the creation, maintenance, layout, navigation and functionality of websites, as well as their compatibility with various devices, browsers and their versions.

### **Some of the responsibilities of front-end developers include:**

One of the responsibilities of front-end developers is the support of various devices and browsers.

Development of dynamic web pages

Creation and maintenance of web pages

Creation and maintenance of applications based on web technologies

Web design

Website styling

Providing convenient navigation and structured content

Design of user interfaces

SEO optimization

Development of solutions based on CMS systems

Maintaining, regardless of browser or device, the same look and functionality



## Technologies used by Front-end Web Development

### HTML and XHTML

HTML and XHTML are the main markup languages for creating web pages and other information to be displayed by a web browser. HTML elements (tags) create the main blocks that make up the structure of the page. It is essential for easier finding of the site by search engines. HTML allows the embedding of images, objects and content. It provides tools for creating structured documents by designating structural semantics for text such as titles, paragraphs, sheets, links, citations, and more. It also allows embedding of scripts written in languages such as Javascript, which in turn affect the behavior of the HTML page.

HTML stands for Hyper Text Markup Language

HTML describes the structure of a web page

HTML consists of a series of elements

HTML elements tell the browser how to display the content

HTML elements are represented by tags

HTML tags label pieces of content such as “title”, “paragraph”, “table”, etc.

Browsers do not display HTML tags, but use them to display the content of the page

### **A simple HTML document**

Example:

```
<!DOCTYPE html >
<html >
<head>
<title>Page Title</title>
</head>
<body>

<h1>My First Heading</h1>
```



`<p>My first paragraph. </p>`

`</body>`

`</html >`

Explained example

The `<!DOCTYPE html>` declaration specifies that this document be HTML5

The `<html>` element is the main element of an HTML page

The `<head>` elements contains meta information about the document

The `<title>` element specifies the title of the document

The `<body>` element contains the content of the visible page

The `<h1>` element defines a large position

The `<p>` element defines a paragraph

## HTML tags

HTML tags are names of elements surrounded by angle brackets:

`<tagname> content goes here ... </tagname>`

HTML tags are usually offered **in pairs** such as `<p>` and `</p>`

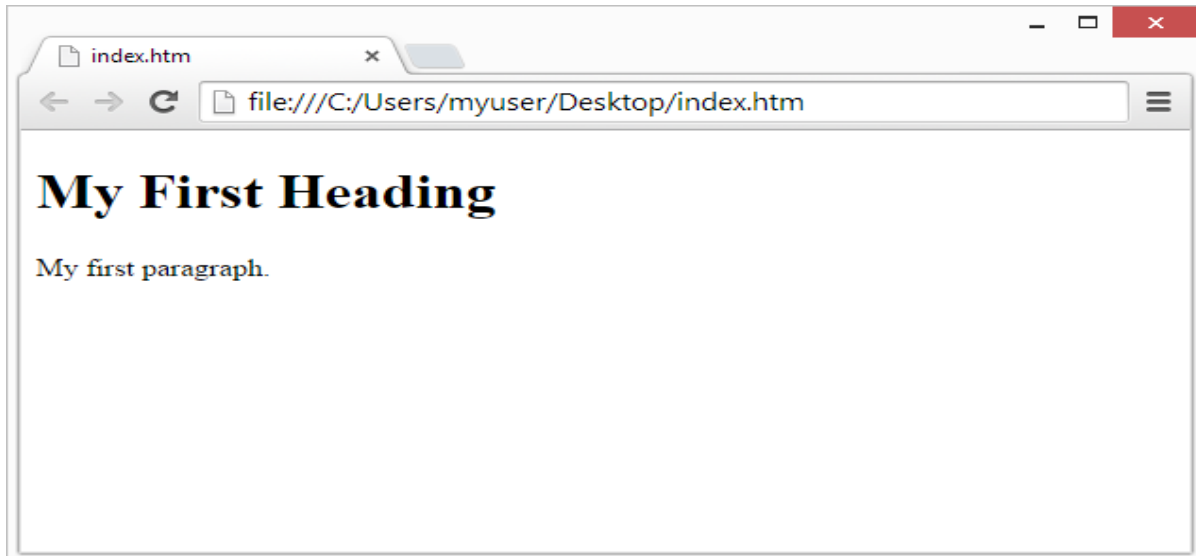
The first tag in a pair is the **start tag**, the second tag is the **end tag**

The end tag is written as the start tag, but with a **leading line** placed before the tag name

## Web browsers

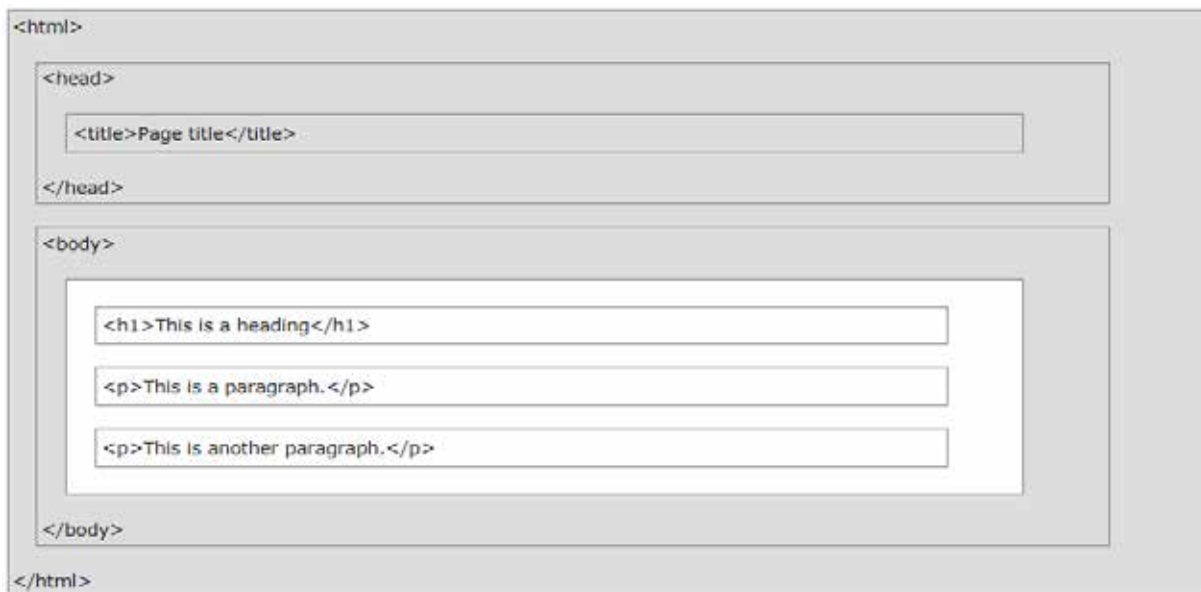
The purpose of the web browser (Chrome, Edge, Firefox, Safari) is to read HTML documents and display them.

The browser does not display HTML tags, but uses them to determine how the document is displayed:



HTML page structure

Below is a preview of the structure of an HTML page:



**Note:** Only the content in the `<body>` section (white area above) is displayed in the browser.

### Declaration `<!DOCTYPE>`

The `<!DOCTYPE>` declaration represents the type of document, and helps browsers to get web pages displayed properly.

It should only appear once, at the top of the page (before all HTML tags).

The `<!DOCTYPE>` declaration is not case sensitive.



The `<!DOCTYPE>`HTML5 declaration is:

```
<!DOCTYPE html>
```

## HTML history

From the early days of the World Wide Web, there are many versions of HTML:

Year	Version
1989	Tim Berners-Lee invented www
1991	Tim Berners-Lee invented HTML
1993	Dave Raggett drafted HTML+
1995	HTML Working Group defined HTML 2.0
1997	W3C Recommendation: HTML 3.2
1999	W3C Recommendation: HTML 4.01
2000	W3C Recommendation: XHTML 1.0
2008	WHATWG HTML5 First Public Draft
2012	<a href="#">WHATWG HTML5 Living Standard</a>
2014	<a href="#">W3C Recommendation: HTML5</a>
2016	W3C Candidate Recommendation: HTML 5.1
2017	<a href="#">W3C Recommendation: HTML5.1 2nd Edition</a>
2017	<a href="#">W3C Recommendation: HTML5.2</a>

## CSS

CSS (Cascading Style Sheets) is a separate language containing many tools that allow you to build the look of HTML pages. Advantages it provides to front-end developers: designed to separate the presentation from the content of the pages, allows easy control over a large number of HTML pages without changing the HTML code of each of the pages, improves the



accessibility of the content and its flexibility, can be applied to any XML document (not to HTML and XHTML only).

CSS stands for C ascading S tyle S heets

CSS describes **how HTML elements should appear on screen, paper, or other media**

CSS **saves a lot of work**. It can control the layout of several web pages at once

External style sheets are stored in CSS files

**CSS Demo - One HTML page - many styles!**

Here we will show an HTML page displayed with four different style sheets. Click on the "Stylesheet 1", "Stylesheet 2", "Stylesheet 3", "Stylesheet 4" links below to see the different styles:

**Why use CSS?**

CSS is used to define styles for your web pages, including design, layout, and display variations for different devices and screen sizes.

CSS example:



```
body {  
  background-color: lightblue;  
}
```

```
h1 {  
  color: white;  
  text-align: center;  
}
```

```
p {  
  font-family: verdana;  
  font-size: 20px;  
}
```

## **CSS has solved a big problem**

HTML has NEVER been intended to contain web page formatting tags!

HTML is designed to **describe the content** of a web page, such as:

```
<h1> This is title </h1>
```

```
<p> This is paragraph. </p>
```

When tags like `<font>` and color attributes were added to the HTML 3.2 specification, it started a nightmare for web developers. Developing large websites, where fonts and color information were added to each page, has become a long and expensive process.

To solve this problem, the World Wide Web Consortium (W3C) created CSS.

CSS removed style formatting from HTML page!

## **CSS saves a lot of work!**

Style definitions are usually saved in external .css files.

With an external style file you can change the look of an entire website by changing just one file!

## **JAVASCRIPT**

JavaScript is an interpreted (scripting), platform-independent programming language that runs on the client. It is most often applied to the HTML of a web page in order to add functionality and load data. JavaScript functions can be associated with page events (for example:





movement/mouse click, keyboard or page element, and other user actions). JavaScript should not be confused with Java.

The main tasks in most JavaScript applications are:

Loading data via AJAX

Effects with images and HTML elements: hide/show, rearrange, drag, slide show, animation and more.

Management of windows and frames.

Recognize browser capabilities.

Using the camera and microphone.

Create 3D graphics WebGL.

JavaScript libraries

A JavaScript library is ready-made code that makes it easier to write applications using pre-created solutions. Examples of JavaScript libraries are jQuery, Prototype, Underscore.js, Sammy.js, Mustache (using and creating templates), Kendo UI (complete development of SPA applications), Modernizr, Angular JS (creating SPA applications), Knockout.js and Ext JS.

### JavaScript can modify HTML content

One of the many HTML methods is `getElementById()`.

This example uses the method to "find" an HTML element (with id = "demonstration") and changes the content of the element ( `innerHTML`) in „Hello, JavaScript“:

Example:

```
document.getElementById("demo").innerHTML = "Hello JavaScript";
```

JavaScript accepts both double and single quotes:

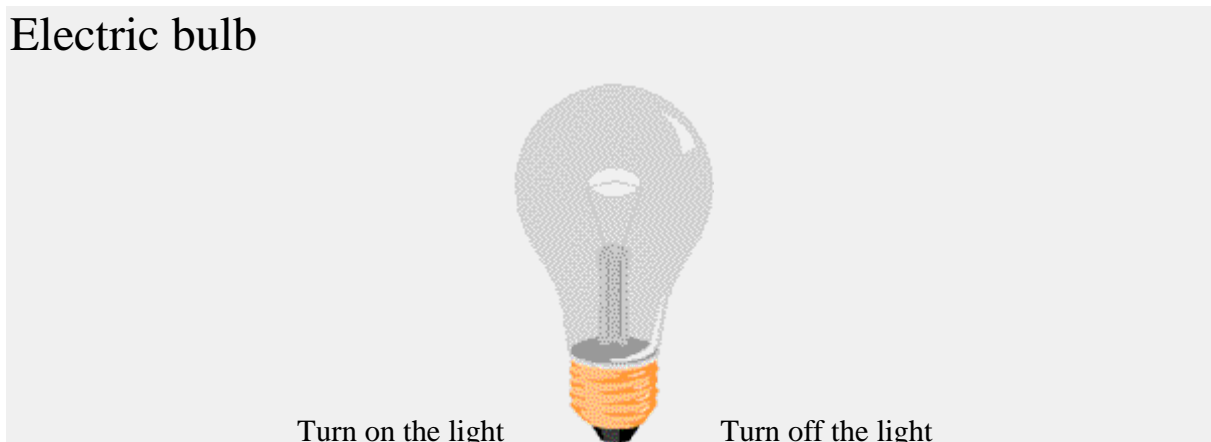
Example:

```
document.getElementById('demo').innerHTML = 'Hello JavaScript';
```

## JavaScript can change the values of HTML attributes

In this example, JavaScript changes the value of the `src` attribute (source) to `<img>` tag:

### Electric bulb



## JavaScript can modify HTML styles (CSS)

Changing the style of an HTML element is a variant of changing the HTML attribute:

Example:

```
document.getElementById("demo").style.fontSize = "35px";
```

## JavaScript can hide HTML elements

Hiding HTML elements can be done by changing the `display` style:

Example:

```
document.getElementById("demo").style.display = "none";
```

## JavaScript can show HTML elements

Displaying hidden HTML elements can also be done by changing the `display` style:

Example:

```
document.getElementById("demo").style.display = "block";
```

## jQuery

The aim of jQuery is to make it easier to use JavaScript on your website.



What you already need to know

Before you start learning jQuery, you should have a basic knowledge of:

HTML

CSS

JavaScript

### **What is jQuery?**

jQuery is a lightweight, "write less, do more" JavaScript library.

The aim of jQuery is to make it easier to use JavaScript on your website.

jQuery takes many common tasks that require many lines of JavaScript code, and turns them into methods that you can call with a single line of code.

jQuery also simplifies many of the complex things of JavaScript, such as AJAX calls and DOM manipulation.

The jQuery library contains the following functions:

HTML / DOM manipulation

CSS manipulation

HTML event methods

Effects and animations

AJAX

utility services

### **Why jQuery?**

There are many other JavaScript libraries, but jQuery is probably the most popular and also the most extensible.

Many of the largest companies on the network use jQuery, such as:

Google

Microsoft



IBM

Netflix

## Will jQuery work in all browsers?

The jQuery team knows all about the problems between browsers and they have recorded this knowledge in the jQuery library. jQuery will work exactly the same in all major browsers.

## Bootstrap 4

Bootstrap is a free front frame for faster and easier web development

Bootstrap includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels, and more, as well as optional JavaScript plugins

Bootstrap also gives you the ability to easily create responsive designs

## What is responsive web design?

Responsive web design means creating websites that automatically tune in to look good on all devices, from small phones to large desktops.

Loading example 4:

```
<div class="jumbotron text-center">  
  <h1>My First Bootstrap Page</h1>  
  <p>Resize this responsive page to see the effect!</p>  
</div>
```

```
<div class="container">  
  <div class="row">  
    <div class="col-sm-4">  
      <h3>Column 1</h3>  
      <p>Lorem ipsum dolor. .</p>  
    </div>  
    <div class="col-sm-4">  
      <h3>Column 2</h3>  
      <p>Lorem ipsum dolor. .</p>  
    </div>  
    <div class="col-sm-4">  
      <h3>Column 3</h3>  
      <p>Lorem ipsum dolor. .</p>  
    </div>  
  </div>  
</div>
```



Bootstrap 4 is the latest version of Bootstrap; with new components, faster sheet styles and more responsiveness.

Bootstrap 4 supports the latest, stable versions of all major browsers and platforms. However, Internet Explorer 9 and below is not supported.

### **Why use Bootstrap?**

Advantages of Bootstrap:

**Easy to use:** Anyone with only a basic knowledge of HTML and CSS can start using Bootstrap

**Sensitive functions:** The **responsive** CSS of Bootstrap adapts to phones, tablets and desktops

**Mobile first approach:** In Bootstrap, styles for the first time for mobile devices are part of the basic framework

**Browser compatibility:** Bootstrap 4 is compatible with all modern browsers (Chrome, Firefox, Internet Explorer 10+, Edge, Safari and Opera)

### **Alternative programming languages**

There are various alternative languages that can compile the final code in JavaScript. They usually add some missing features and implement their own language structures. Such languages are Coffeescript, Kaffeine and Dart.[5]

### **CMS**

The great popularity of CMS programs (or Content Management Systems), which is due to a number of facts (including the ease of managing the content of the site) requires the work of front-end developers with them. CMS systems allow end users to quickly and easily, without knowledge of HTML, edit the content of their sites: creating new ones, deleting old ones and editing existing articles, adding menus, correcting hyperlinks, installing additional components, etc.). Some of the most popular CMS programs are: WordPress, Drupal, Joomla! and Sitefinity.[6]



### Other Web technologies

Adobe Flash is one of the most popular and used multimedia environments for creating animated websites, web ads, web games, as well as various programs and presentations. There are rich possibilities for video and audio reproduction, which makes the environment extremely attractive, as well as a major tool in the work of many professional designers. Flash allows the creation of so-called rich internet applications.

Microsoft Silverlight is a technology for creating web applications with similar functions and purposes to applications created with Flash. Silverlight is most commonly used for video streaming, multimedia, computer graphics and animation.

### **3. Back-End Web Development: Java, MySQL**

The back-end application serves as support for front-end services. It can interact directly with the front-end application or, more commonly in practice, with a program called (intermediate program), which mediates between the operations of front and back-end applications.

The back-end developer takes care of the internal infrastructure of the site - the way the site works, security, content management, updates, database, server, etc. He knows PHP, C++, Java, .Net and other languages. Usually a good back-end programmer knows one of these languages and is familiar enough to write code in a second or third language. A prerequisite for them is to work with databases such as MySQL, Oracle and SQL Server.

Their task is to build the site so that if the traffic to it increases sharply, it will not stop working properly. In addition, the thought of the back-end programmer runs in the direction - if you need to add new functionality, the site should be made so that adding new functionality does not break everything else. Each link and each process that takes place on the site is well thought out in advance so that later the site administrator can easily upload new content, add new buttons to the menu and change existing content without harming the site.



Back-end developers mainly develop and maintain the functional logic and operations of a software application or information system. Typically, a back-end programmer has expert programming skills in C++, C #, Java, and other high-level programming languages.

The key role of the back-end developer's work is to ensure that the data or services requested by the front-end system or software are delivered through software. Back-end developers also create and maintain the entire back-end of the system, which consists of the basic logic of the application, databases, data and application integration, API and other back-end processes. In addition, a back-end programmer performs testing and debugging of any back-end application or system.

The responsibilities of the back-end developer may include working with:

Creating, integrating and managing databases

Backup frameworks for server software development

Web server technologies

Integration in cloud computers

Server programming languages

Operational system

Development, implementation and maintenance of a content management system

API integration

Security settings and hacking prevention

Reporting - generating analyzes and statistics

Technologies for archiving and restoring files and database of a website





Back-end developers typically work with languages such as Python, Ruby, PHP, Go, or server JavaScript. As the technology that powers the network is based on open standards, there are many options for choosing which language the back-end developer should choose.

## **Java**

Java is a popular programming language created in 1995.

It is owned by Oracle and over **3 billion devices** run Java.

### **It is used for:**

Mobile apps (Android apps in particular)

Desktop applications

Web applications

Web servers and application servers

Games

Database connection

And many, many more!

### **Why use Java?**

Java runs on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.)

It is one of the most popular programming languages in the world

It is easy to learn and easy to use

It is open source and free

It is safe, fast and powerful

There is huge support from the community (tens of millions of developers)

Java is an object-oriented language that provides a clear structure of programs and allows code reuse, reducing development costs

As Java is close to C ++ and C # , this makes it easy for programmers to switch to Java or vice versa



## SQL

SQL is a standard language for accessing and manipulating databases.

SQL stands for Structured Query Language

SQL allows you to access and manipulate databases

SQL became a standard of the American National Standardization Institute (ANSI) in 1986 and of the International Standardization Organization (ISO) in 1987.

### **What can SQL do?**

SQL can execute database queries

SQL can extract data from a database

SQL can insert records into a database

SQL can update records in a database

SQL can delete records from a database

SQL can create new databases

SQL can create new tables in a database

SQL can create stored procedures in a database

SQL can create database views

SQL can set permissions for tables, procedures, and views

### **SQL is a standard - BUT ....**

Although SQL is an ANSI/ISO standard, there are different versions of the SQL language.

However, to be compatible with the ANSI standard, they all support at least the basic commands (such as SELECT, UPDATE, DELETE, INSERT, WHERE) in a similar way.

**Note:** Most SQL database programs also have their own extensions in addition to the SQL standard!

### **Using SQL on your website**

To create a website that displays database data, you will need:



RDBMS database program (i.e. MS Access, SQL Server, MySQL)

To use a server scripting language, such as PHP or ASP

To use SQL to get the data you want

To use HTML/CSS to format the page

## **RDBMS**

RDBMS stands for Relational Database Management System.

RDBMS is the basis for SQL and for all modern database systems such as MS SQL Server, IBM DB2, Oracle, MySQL and Microsoft Access.

RDBMS data is stored in database objects called tables. A table is a collection of related data records and consists of columns and rows.

See “Customers” table:

Example:

```
SELECT * FROM Customers;
```

Each table is broken down into smaller units called fields. The fields in the customer table consist of CustomerID, CustomerName, ContactName, Address, City, PostalCode, and country. A field is a column in a table that is designed to support specific information for each record in the table.

A record, also called a row, is each individual record that exists in a table. For example, there are 91 records in the customer table above. The record is a horizontal formation in a table.

A column is a vertical unit in a table that contains all the information associated with a particular field in a table.

## **4. AJAX Angular.js programming**

### **AJAX**

#### **What is AJAX?**

AJAX = A synchronous JavaScript and XML.

AJAX is not a programming language.

AJAX simply uses a combination of:

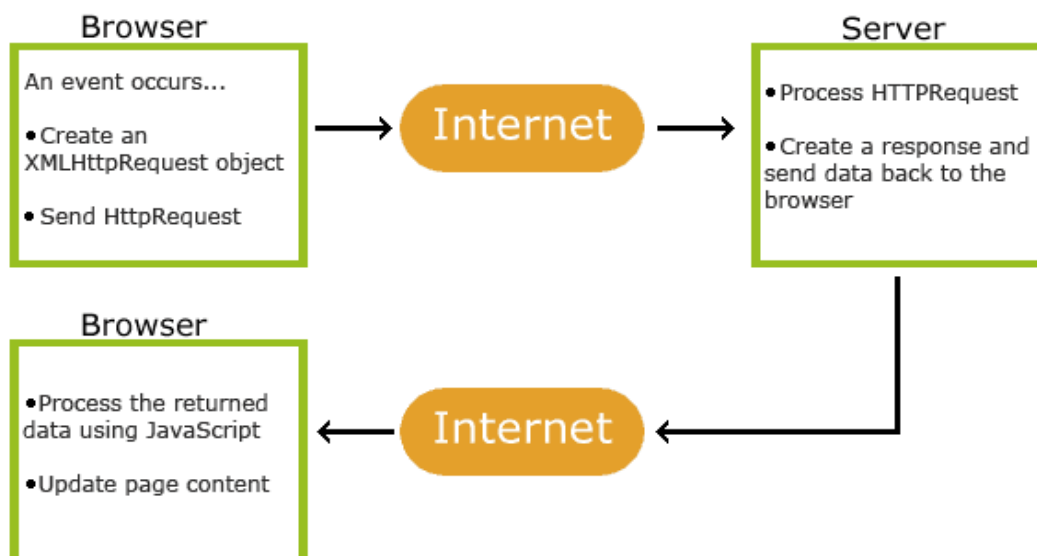
Embedded XMLHttpRequest object in the browser (for requesting data from a web server)

JavaScript and HTML DOM (to display or use data)

AJAX is a misleading name. AJAX applications can use XML to transport data, but this is just as common for transporting data as plain text or JSON text.

AJAX allows web pages to be updated asynchronously by exchanging data with a web server behind the scenes. This means that it is possible to update parts of a web page without reloading the entire page.

### How AJAX works



1. Event occurs on a web page (page is loaded, button is pressed)
2. An XMLHttpRequest object is created from JavaScript
3. The XMLHttpRequest object sends a request to a web server



4. The server is processing the request
5. The server sends a response back to the web page
6. The response is read by JavaScript
7. The correct action (such as a page update) is done by JavaScript

## **Angular**

“Angular” is an open source platform for web applications. Developed by Misko Hevery and Adam Abrons in 2009, it is supported by Google and a community of developers and corporations looking to deal with various difficulties in developing single-page applications (or SPA). The aim is to simplify both the development and testing of this type of application by providing a platform based on the architecture model - Model-View-Controller, along with all commonly used components for Internet applications.

## **Philosophy**

AngularJS has been developed following the idea that declarative programming should be used to develop user software and linking software components, while imperative programming is better used in defining business logic in the application. The platform adapts and increases the capabilities of traditional HTML to present dynamic content through two-way data collection (data-binding), which allows automatic synchronization of model and view (MV). As a result, AngularJS does not place so much emphasis on DOM manipulation, but aims to improve the ability to test the application and its performance.

## **Basic concepts in AngularJS**

Template - HTML with additional tagging

Directives - extend HTML with additional user attributes and elements

Model - the data that the user sees in the View and with which he exchanges information

Scope - the context where the model is stored so that controllers, directives and expressions can access it

Expressions - access the variables and functions of the scope

Compiler - parses the template, directives and expressions

Filter - formats the value of the end user expressions

View - what the user sees (DOM)

Data Binding - synchronizes data between the Model and the View

Controller - the business logic behind the View

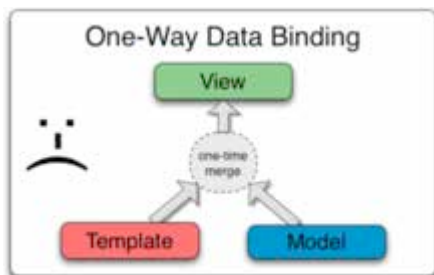
Dependency Injection - creates and connects objects and functions

Injector - the container that stores Dependency Injection

Service - business logic that is used repeatedly (regardless of the View)

Module - a container for the various parts of the application, includes controllers, services, filters, directives that configure the Injector

## Data Binding

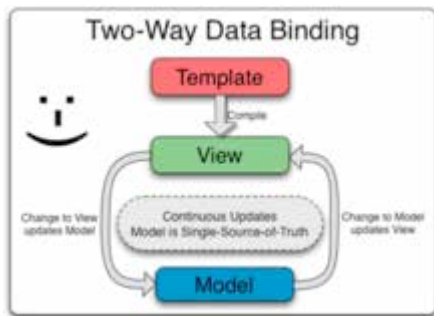


Data Binding in AngularJS applications is an automatic synchronization of data between the model and the view components. The way AngularJS performs data binding allows the model to be considered as a source-of-truth in the application. The view is a projection of the model at any time. When the model changes the view reflects the change, and vice versa.

## Data Binding in Classical Template Systems

Most template systems connect data in only one direction: i.e. they merge the template and the model together in the view. After a merger occurs, changes to the model or its corresponding sections are not reflected in the view. Even worse, not all changes the user makes to the view are reflected in the model. This means that the developer must write the code to constantly synchronize the view with the model and the model with the view.

## Data Binding in Angular Templates



AngularJS templates work differently. The template is first compiled in the browser. The compilation stage produces the live view. All changes to the view are immediately reflected in the model, and the data from the model is immediately reflected in the view. The model is a source-of-truth about the state of the application, a greatly simplified software model for the developer. One can think of the view simply as a momentary projection of the model.

As the view is simply a projection of the model, the controller is completely separate from the view. This makes testing the controller quick and easy due to its isolation from the view and the associated DOM/browser.

### Directives

At a high level, directives are tags in DOM elements (similar to attributes, element names, comments, or CSS classes) that allow the compiler (`$compile`) to add specific instructions and even convert the DOM element and its children. Angular has embedded directives such as `ng-Bind`, `ng-Model` and `ng-Class`. It is assumed to start with the prefix `ng-`. It is possible to create your own directives to be used.

### Creation of Directives

It is preferable to use directives by tag names instead of comment or class names. This makes it easier to determine what a specific directive aims to do.





The directives are organized in modules. The module.directive API is used to create a new directive. This API consists of a directive name, as well as an embedded function that returns an object showing the compiler how the directive behaves when used.

The compiler can use directives, calling them by element names, attributes, class names, and comments.

Sample directives:

ng-app stands for AngularJS application

ng-init uses the adjacent data in the application

ng-model is used to link text data, selections and HTML input to the AngularJS application

ng-repeat repeats HTML elements once for each key in an array

## Angular SEED

Angular SEED is a ready-made template that improves the efficiency and speed of web application development, especially when using CSS, Node JS and Unit Testing. It is applicable mainly for small projects. It can be downloaded from the official AngularJS website.

## “Hello World” Program

Below you can see an example of the "Hello World" program in AngularJS:

In Default.html:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hello, World!</title>
    <script
src="http://ajax.googleapis.com/ajax/libs/angularjs/1.0.7/angular.min.js"><
  /script>
  <script src="controller.js"></script>
  </head>
  <body>
    <div ng-app="helloWorldModule">
    <div ng-controller="HelloWorldController">
Name: <input type="text" ng-model="name" required>
    <hr>
```



```
<div>Hello, {{name}}!</div>
</div>
</div>
</body>
</html>
```

In `controller.js`:

```
'use strict';
// define the module
var helloWorldModule = angular.module("helloWorldModule", []);

// add module controller
helloWorldModule.controller("HelloWorldController", function ($scope) {
  $scope.name = "World";
});
```

## 5. Website development and modification via CMS

Content Management System (CMS) is a type of application software that allows the publication and editing of a website, as well as support for the main interface. The aim is to facilitate the development of a dynamic website, with the ability to easily and quickly change its content. These systems greatly facilitate teamwork and offer many opportunities to delegate various administrative rights and roles in the process of creating and editing content.

CMS content management systems are web applications that allow website administrators to easily publish, supplement, edit and delete content, photos and documentation on their site.

A web application needs web hosting and a domain name in order to exist on the Internet.

Many websites are built entirely on these web-based applications, as they do not require in-depth knowledge of HTML and programming languages. CMS programs make website administration accessible to anyone who wants to develop their own website, even if they have no experience in this field.

Content management systems are often used in a variety of websites: corporate or personal blogs, news portals and e-stores. This technology aims to avoid or at least minimize the need for manual code writing. This significantly reduces the need for specialists to maintain the



program code. This division allows system users to focus on creating and editing the content itself, rather than on the technologies used, so that it can be made available to the target audience.

### **Main characteristics**

The main purpose of content management systems is to manage information on websites and corporate systems. Whether it's a personal blog or a corporate website, the content management system allows users without knowledge of HTML and CSS to add content. There is an extremely large selection of such systems and each of them is different, but the common is that they have an intuitive interface and can be used by people without programming skills. Some of these systems provide only the functions necessary for the average user in an easy and accessible way, and others - the possibility of extremely complex data manipulation. Most management systems include publishing, formatting, reviewing, indexing, searching, and retrieving text. The system can be used as a central repository containing documents, movies, photos, phone numbers, scientific data and more. It can be used to store, control, review, semantically enrich and publish documentation. Users can easily add and delete pictures and photos and edit the text on their own websites. The system automatically generates hyperlinks and takes care of users' access rights.

### **Component content management system**

The component content management system specializes in creating component documents. For example, CCMS, which uses DITA XML, allows users to assemble individual themes into one card (document). These components can be reused with another document or with a plurality of others. This ensures that the content is monitored throughout the documentation set. However, this type of system is not suitable for large organizations because they have the opportunity to order a system that meets their requirements.

### **Corporate content management system**

The corporate content management system organizes documents, contacts and records related to the processes of the commercial organization. This system structures corporate information



in a way that is most appropriate for the organization, providing easy access to it by employees and customers and ensuring maximum information security.

### **Digital asset management systems**

Digital Asset Management (DAM) consists of tasks and solutions related to cataloging, storing, annotating, retrieving and distributing digital assets. Digital photos, animations, videos and music files represent the target areas of media asset management (subcategory of the DAM). Digital Asset Management Systems (DAMS) contain software and hardware systems that support the digital asset management process. The term "digital asset management" also refers to the protocol for downloading, renaming, copying, grouping, archiving, optimizing, maintaining and exporting files.

The managed "asset" is taken and stored in digital format. There is usually a target version, or also called a "real" version of the asset with the highest resolution and most reliable reproduction. The asset is described through metadata. Metadata is the description of the asset itself, and the depth of the description may vary depending on the needs of the system, creator or user.

Metadata describes:

- The content of the asset (what is contained in the package);
- Decoding/encoding method (example: JPEG, tar, MPEG 2);
- Origin - the history of the asset until the moment of "capture";
- Asset ownership;
- Access right;
- Other.

### **How do CMS platforms work?**

CMS web platforms create a dynamic environment in which all content is written to a specific database or XML files.



Using a lightweight and accessible interface, the administrator can choose which page of the website to edit and effortlessly make the necessary changes or new content.

Once the content is updated, with the touch of a button, the CMS system will convert the text into HTML code and publish it on your website.

One of the most important aspects of a good CMS program is the ease of use and the many content processing options. The main advantage of these systems is that you do not need to know programming languages. These programs provide you with a text editor that has the same features and capabilities as standard word processing programs such as Microsoft Word or Open Office. In this way you can easily prepare the texts for your website, and the CMS system will generate the necessary HTML code and publish them on the page of your choice. Many CMS programs allow the integration of an On-line store, forum or chat, through which you can easily expand the functionality of your website.

### **What types of CMS exist?**

There are different types of systems that differ mainly in the level of functionality. Some of them contain extremely powerful process management tools, but in return require more in-depth knowledge from users. Such systems are suitable for large organizations that have sufficient resources to maintain them. Other systems rely on the most simple and intuitive user interface and allow even people with minimal computer knowledge and skills to share content.

There are dozens and probably hundreds of such systems. Some allow the site to do virtually anything, but this is usually at the expense of excessive complexity and difficult maintenance. Few customers can handle such systems without in-depth training and experience.

There are also other types of systems which functionalities are relatively limited to what is needed for regular and trouble-free updating of the site. These systems are much lighter and simpler to maintain and much more intuitive to operate.

### **Popularity of content management systems**



About 50% of all websites on the Internet use some form of content management system. WordPress has the highest share, which drives more than half of websites with such systems. It is followed by Joomla, Drupal, modX, textPattern, Refinery CMS, Concrete5, DotNetNuke, Umbraco, ExpressionEngine, Radiant, SilverStripe, Alfresco, Contao (TYPOLight).

## **6. Development environments: Integrated debugging environment (IDE), development tools, tools to support methodological control of sources, error tracking, automated tools for implementation/continuous integration, etc.**

The integrated development environment is a software application that provides a complete software development environment to programmers.

An IDE usually consists of a code editor, tools to automate the development of the source application, and a debugger. Most modern integrated environments offer automated code prediction and completion.

Some IDEs include a compiler, interpreter, or both, such as NetBeans and Eclipse; others do not have this functionality, such as SharpDevelop and Lazarus. There is no well-defined boundary between an integrated development environment and another type of software development tool. In some cases, the package includes a version control system or various graphical interface design tools. Many modern development environments also have a glossary of classes and a diagram with their hierarchy, to facilitate object-oriented programming.

Integrated development environments aim to allow the programmer to achieve maximum productivity by providing closely related components with a similar user interface. In contrast to the use of various software products such as vi, GCC or make, IDE is an application that performs all the processes associated with software development - writing code, editing, compiling, publishing and debugging.



One of the goals of the IDE is to reduce the configuration required to share multiple development tools, instead offering the same set of capabilities in a cohesive unit. Reducing organization time can increase user productivity if learning to work with the new environment is faster than manually connecting individual applications. The integrated development environment can also help with other means, for example the written code can be analyzed at the time of introduction and potential syntax errors can be identified in real time. This can help with learning a new programming language or library.

Some IDEs specialize in a particular programming language, including tools that most closely match its programming paradigms. Others support many languages and can be extended with additional modules at the user's request, such as Eclipse, IntelliJ IDEA, Oracle JDeveloper, NetBeans and Microsoft Visual Studio.

Although most modern integrated environments have a graphical interface, applications such as Turbo Pascal, which is entirely text-based, were popular before the widespread introduction of windowed operating systems such as Microsoft Windows or X Window System (X11). With them, frequently used functions are called up with keyboard shortcuts.

### **Visual programming**

Visual programming usually requires an integrated development environment. Visual IDEs allow users to create new applications by shifting program blocks or code links to produce flow or block diagrams, which are then compiled or interpreted. These graphs are often based on a unified modeling language.

This type of interface is gaining popularity through the Lego Mindstorms system and is actively sought after by a number of companies that want to take advantage of the power of specialized browsers, such as Mozilla. KTechlab supports flowcharts and is a popular open source IDE and Simulator for developing microcontroller software. Visual programming is primarily responsible for the power of Distributed Computing Systems (see LabView and EICASLAB software). The early Max visual programming system was modeled on an analog synthesizer and has been used since 1980 to develop software for real-time music performances. Another early example is Prograph, a system based on data



streams originally developed for Macintosh computers. The Grape graphical environment is used to program qfix robotics packages.

This approach is also used in specialized software such as Openlab, where end users want full flexibility in a programming language without the difficulty of learning one.

### **Language support**

Some IDEs support multiple languages, such as GNU Emacs, based on the „C“ language and EmacsLisp. IntelliJ IDEA, Eclipse, MyEclipse or NetBeans, all based on Java language. Also MonoDevelop, based on C#.

The support of alternative languages is often done through plug-ins, that allow them to be installed on the same IDE at the same time. For example, Flycheck is a state-of-the-art real-time syntax checking system for GNU Emacs 24 that supports 39 languages.[3] Eclipse and NetBeans have plug-ins for C/C++, Ada, GINAD (for example, Ada Gide), Perl, Python, Ruby and PHP, that are activated automatically based on file extension, environment, or project settings.

### **User attitude**

UNIX programmers can combine POSIX command-line tools in a complete development environment that allows large programs such as the Linux kernel and its environment to be developed.[4] The free software tools GNU (GNU Compiler Collection (GCC), GNU Debugger (gdb) and GNU make) are available on many platforms, including Windows.[5] Developers who prefer the command-line can use editors that support many of the standard UNIX and GNU-developed tools to develop their own IDE with programs such as Emacs[6][7][8] or Vim. Display Data Debugger is designed to be an advanced graphical front-end of many text based standard debugger tools. Some programmers prefer to work with make-files instead of similar tools included in a full IDE. For example, most participants in PostgreSQL database development directly use make and gdb to develop new functions.[9] Even when developing PostgreSQL for Microsoft Windows using Visual C++, Perl scripts are used as a substitute for make, rather than relying on an integrated development





environment.[10] Some Linux IDEs, such as Geany, try to provide a graphical front-end to the traditional development operations.

On various Microsoft Windows platforms, command-line development tools are rarely used. Accordingly, there are many commercial and free products that have different designs, often creating incompatibilities. The largest providers of Windows compilers also provide free copies of their command-line tools, including Microsoft (Visual C ++, Platform SDK, .NET Framework SDK), Embarcadero Technologies (bcc32 compiler)

IDEs have always been popular on Apple Macintosh Mac OS, dating back to the Macintosh Programmer's Workshop, Turbo Pascal, THINK Pascal and THINK C circles, from the mid-1980s. Mac OS X programmers can choose between local IDEs such as Xcode and open source tools such as Eclipse and Netbeans. ActiveState Komodo is a patented IDE supported by Mac OS. With the advent of cloud services some development environments are available on-line and run within web browsers.

## **7. Databases: development of basic skills of students in analyzing and applying appropriate models for data processing**

Databases are used everywhere on the Internet to store a wide variety of information: on-line dictionaries and encyclopedias, library catalogs, bank depositors' account data, millions of search engine indexed websites, catalogs of on-line trading companies and much more.

What makes databases so popular? The information in the databases is arranged in a certain way, which makes its computer processing possible. For example, the most common relational databases on the Internet arrange the information entered in them in the form of tables. Each table has rows, columns, and row numbers that are very similar to an Excel spreadsheet. In fact, many relational databases allow direct input of data from Excel spreadsheets.

A slightly more specific example of relational databases are the tables with names and passwords of users registered to a site. If you are entering a site for the first time that requires



registration, you are expected to fill in a form with personal data, username and password. This data is stored in the database as a row of a table containing information about all registered users.

## **Database management systems**

In order for databases to be useful, a way is needed to process the information stored in them. This task is performed by computer programs - the so-called database management systems. There are a large number of such programs: MySQL, Access, Oracle, dBase, Microsoft SQL Server, etc. Each has its own characteristics, but in general they all perform basic functions: creating, processing, editing and deleting data from the database.

One of the most popular database management systems is MySQL. The program is very stable, secure and able to maintain huge databases, such as the accounts of millions of Facebook users or Google indexes. In addition, MySQL is free to use and widespread: if your hosting company offers you the use of databases, MySQL is almost certainly among the options offered.

## **Databases and PHP**

MySQL can be managed directly with SQL commands issued, for example, through the Internet-based MySQL administration tool phpMyAdmin. Another very popular way to manage MySQL is to use the PHP programming language. The MySQL-PHP combination can be behind such applications as a guestbook, the already mentioned registration on websites, Facebook or popular content management systems (CMS) such as Joomla! and Drupal.

PHP allows you to dynamically manage the contents of the database. For example, let your site contain encyclopedic articles on sports topics. If you don't feel like an expert in all sports, you can leave it up to users to edit what you've already written or add new articles (as you can imagine, it's something like Wikipedia). You can do this with an appropriate PHP program that instructs users to make adjustments to the sports article database.



In addition to encyclopedic articles (Wikipedia), users could add video (YouTube) or news (Digg). Thanks to the capabilities provided by PHP and MySQL, your site can grow, be dynamic, in short, enjoy all the benefits that a collective work has.

In conclusion, databases and tools for their management are one of your best helpers in developing and promoting your site.

## **8. Interface design: development of students' competencies for entering complex scenarios for use and independent participation in the design process related to the design of complex user interfaces**

Basic principles in the development of interfaces. The principles of design development are based on a number of developments and research done by Galitz (1992), IBM (1991, 2001), Mayhew (1992), Microsoft (1992, 1995, 2001, 2010), the Open Source Software Foundation (1993), as well as over the years of our practice.

The interface must be aesthetically pleasing, providing a good and expressive contrast between the screen elements.

1. Grouping of functionally distinct elements;
2. Aligning all design elements and groups. Observing the rule for consistent margins with odd values.

The use of color scheme and graphics should be moderate and simple, without straining the eye. The messages they send must be clear and quick to understand. An incorrect composition can confuse the user and have a disorienting effect.

**Transparency.** The interface must be visually, conceptually and linguistically clear and understandable, including:

1. Visual elements;
2. Functions;



3. Metaphors or analogies we use;

4. Words and texts.

**Coherency.** The interface is coherent if there is:

1. Coherency with user needs: the design must meet the requirements of the customers. An effective interface starts with realizing what the user needs, how he sees things. One of the common mistakes made by designers is that they consider all customers to be the same. Another common mistake that is made - the user thinks, feels, acts like a programmer, but this is not the case. Our customers have their own understandings and preferences. If an interface reflects only the views of the designer, it will not be successful. "Know the customer" is a fundamental principle in this type.

2. Coherency with the tasks: The organization of the system must correspond to the work to be performed. The structure of the functional flow must allow easy passage through the tasks. The screens that the user has to go through, especially for frequently repeated tasks, should be minimized.

3. Coherency with the versions of a product / site: The user of a new system has often used similar ones before. The habits and expectations he has are to be compatible (at least partially for such actions). This way you will not have to learn everything again and get bored or "scared" of innovations.

**Accessibility.** The interface must allow easy learning and understanding of the system.

The user should easily recognize the following:

1. What to do?

2. When to do it?

3. Where to do it?

4. Why do it?



## 5. How to do it?

The sequence of actions, answers, visual representations and information must be carefully selected and placed in the right context.

**Configuration.** Enable easy customization, configuration and reconfiguration of settings. Easy customization by configuring settings gives the user a sense of greater control over the system. They allow him to participate more actively and thus learn it more easily. Of course, for users who are not fans of many settings, we must provide default settings.

**Consistency.** Make the system work consistently. Such components must:

1. Have a similar visualization;
2. Have a similar use;
3. Operate in a similar way;
4. Same action must always return the same result;
5. The function of the elements must not be changed;
6. The position of the standard elements must be the same on all screens/pages.

**Control.** The user must control the interaction:

1. Actions must work with every explicit request from the user;
2. Actions must be carried out without delay;
3. Actions must be able to be undone by the user.

**Management.** Tasks must be performed directly. The available options should be easily visible.

**Expediency.** Minimize hand and eye movements:



1. The transitions between the different control systems must be smooth and easy;
2. The path to a function should be as short as possible;
3. The movement of the eye through the screens must be one-way and consistent, and not jump over the screen;
4. Anticipate the desires and needs of users whenever possible.

**Awareness.** Develop the interface on the basis of already existing user knowledge, on interface concepts, terminology familiar to the customer. Operations are good to emulate the same behavioral patterns. Familiar concepts allow people to get started and become productive quickly.

**Flexibility.** The system must be sensitive to the different needs of its users. In this way it will be able to meet their individual requirements.

**Indulgence.** Tolerate and forgive the usual and inevitable human mistakes.

1. Prevent error from occurring whenever possible;
2. Protect yourself from possible system failures;
3. If an error occurs, provide a "talking" message.

**Predictability.** Tasks, screens and movements through the system must be predictable based on previous experience and knowledge. All actions must lead to the result expected by the user. The elements of the interface must be distinguishable and easily recognizable. Current operations should suggest what will happen next.

**Recovery.** The user must be able to reverse the action with the "cancel" command. Knowing that you can cancel an operation saves a lot of worries, especially for new users. The goal is greater system stability or easy reversal of a wrong action in the application.



**Response.** The system must respond immediately to a request from the user. Provide a quick response for any user action:

1. Visually;
2. Text;
3. Multimedia (if possible).

Never leave the screen blank for more than a moment, because the user may think that the system has failed. You can write text, put an animation showing that an operation is currently in progress.

**Simplicity.** Try to keep your interface as simple as possible. Some of the ways to achieve this are:

1. Use progressive exposure - hide and show parts of the interface only when needed;
2. Introduce first only the necessary and usual functions;
3. Hide less used and complex functionalities;
4. Always provide default values;
5. Make common actions simple.

**Compromises.** The final design is always the result of a number of compromises. They lead to another cardinal rule in graphic design: Customer requirements usually take precedence over technical conditions. It is always easier for the designer not to comply with the user, but then the final interface will not meet his needs. In order to achieve maximum results, we must know all the principles well and decide carefully what we can compromise with and to what extent. This development and decision-making process can be divided into several steps, described below.



## **Steps in the interface design process.**

### ***Step 1: Getting to know the user or the customer***

When we start designing the interface, we must get acquainted with the most important part of the computer system: our user. He is the one who will use our product. This research is difficult and often underestimated as an important factor, but is critical to the success of our software or website.

### ***Step 2: Defining the Business functions and the type of the application***

In this step, the business functions that the future software will perform are studied and formulated.

What are the requirements? The tasks that will be performed by the users are analyzed and described.

The design standards, which will be observed and the type of application are determined. Based on them, the conceptual model of the system is defined later.

## **Features of some basic types of programs:**

### **1. Office, industry applications:**

Target users: *employees in scientific and administrative environment.*

Objectives: *productivity, cost reduction.*

Use: *frequent use for a long period, e.g. throughout the working day.*

User expectations: *familiarity, consistency, fast productivity.*

Suitable use of WPF technology: *yes.*

### **2. Customer applications**





Target users: *customers*.

Objectives: *for users performing specific tasks*.

Use: *use for a short period (e.g. weekly)*.

User expectations: *easy use and good performance of the tasks assigned to the system*.

Suitable use of WPF technology: *yes*.

Examples: *multimedia applications, media players, data protection tools*.

### **3. Games**

Target users: *gamers*.

Objectives: *for fun*.

Use: *from time to time, for efficiency*.

Suitable use of WPF technology: *yes, possibility for easy and attractive customization of the interface as well as interactivity*.

Examples: *board and Internet games*.

Some basic ways at this stage to improve the UX design of the application: *model the real world; show instead of explain - use interactive techniques instead of text;*

*improve the meaning of the objects used in the application - if it is a text field in a desktop application, put a label, if it is a picture on the web, put a "title" and "alt" tag; use intuitive icons; reduce the need for prior knowledge of users about such systems; improve feedback tools; make objects easier to work with; focus users' attention on important design elements.*

### ***Step 3: Defining the principles of good computer design***



Here follows the compilation of basic principles described above, which we must follow in interface design. Discussion begins with the customer of the various parts and elements, navigation, screens, typography, reading, browsing, searching the Web (if it is a site/web application).

***Step 4: Development of system menus and navigation schemes***

Whether it is a desktop or web application, it contains some information and performs some functions. In this step, we need to provide users with some mechanism to tell them exactly what they can use our system for. This can be done through lists of options and/or selection of a specific action of a given step, presented by menus.

***Step 5: Selection of appropriate windows (Windows)***

***Step 6: Selection of appropriate input and output controls***

This defines the mechanism through which the user interacts with the application. In recent years, the graphics system has undergone rapid development and now has a whole new family of controls and devices that support communication between user and application.

***Step 7: Providing a system for effective feedback and support***

***Step 8: Creating meaningful and easily recognizable graphics, icons, pictures, multimedia***

***Step 9: Selection of appropriate colors/color schemes***

In recent years, due to the development of hardware and technology, it has become possible to use colors more efficiently in screen design. Previously, it was possible to visualize only a few colors. However, this evolution does not eliminate some of the color problems. Therefore, it is good to consider the following recommendations when choosing color schemes for applications in:

1. Interface for statistical programs: no more than six colors at a time.
2. Web page interface: the palette with 216 colors [7].



### *Step 10: Organizing and arranging windows/pages*

This positioning is very important firstly because of the fast and accurate determination of where the information is and how we can use it, and secondly because of the implementation of the application. For example, if we have a software in which many panels are embedded, this can lead to its malfunction or even failure. It is good to observe the alignment and balance of controls for better visual perception.

Also avoid using odd values for margins, for example 25 instead of 22.

### *Step 11: Testing*

Testing is a very important element for interface design. It allows us at an early stage to detect inaccuracies, misalignments, overlaps of elements that could lead to loss of functionality, and hence users. For a desktop product, this can be incorrectly positioned controls, for a Web application - misspelled CSS styles, embedded animations, leading to visualization problems under different browsers.

**Conclusion.** Effective GUI design for Web and desktop applications is a complex process that goes through a number of steps. The use of "good" interfaces in the construction of Business Information Systems is essential. Many principles, limitations and factors must be taken into account in its development. When it comes to desktop software, these are the many design elements, the types of screens/windows that are used, the way the elements are organized, which controls are selected to extract or process certain data, compatibility with operating systems and databases. If it is Web design, the factors include choosing the appropriate integration of text, graphics, hyperlinks, Web page sizes, multimedia, browser features, and data monitoring.



## PRACTICAL EXERCISES AND TESTS

### EXERCISES

#### HTML

1. Add a "hint" to the paragraph below with the text "All About W3Schools".

```
<p ="About W3Schools">W3Schools is a web developer's  
site. </p>
```

2. Set the image size to 250 pixels wide and 400 pixels high.

```

```

3. Use the correct HTML tag to add a title „London“.

```
<p>London is the capital city of England. It is the most  
populous city in the United Kingdom, with a metropolitan area  
of over 13 million inhabitants. </p>
```

4. Add a horizontal rule between the title and paragraph.

```
<h1>London</h1>
```

```
<p>London is the capital city of England. It is the most  
populous city in the United Kingdom, with a metropolitan area  
of over 13 million inhabitants. </p>
```



5. Use the correct HTML tag to add a paragraph with the text "Hello World!".

```
<html >  
<body>  
  
</body>  
</html >
```

6. Clean this document with appropriate end tags.

```
<h1>This is a Heading   
<p>This is a paragraph. 
```

7. Use the correct HTML attribute and CSS to set the paragraph color to "blue".

```
<p  = "; ">This is a paragraph. </p>
```

8. Use CSS to set the paragraph font to "courier".

```
<p style = " : courier; ">This is a paragraph. </p>
```

9. Use CSS to align the center of the paragraph.

```
<p style = " : center; ">This is a paragraph. </p>
```

10. Add an additional meaning to the word "degradation" in the paragraph below.

```
<p>  
WWF's mission is to stop the  degradation  of our  
planet's natural environment.  
</p>
```



11. Highlight the word "FUN" in the text below.

```
<p>  
HTML is  FUN  to learn!  
</p>
```

12. Add a line by crossing out the letters "blue" in the text below.

```
<p>  
My favorite color is  blue  red.  
</p>
```

13. Use an HTML element to add quotes around the letters "cool".

```
<p>  
I am so  cool .  
</p>
```

14. Make the text below go from right to left.

```
 What a beautiful day! 
```

15. Use CSS to set the document font to "courier".

```
<!DOCTYPE html >  
<html >  
<head>  
<style>  
body {  : couri er; }  
</styl e>  
</head>  
<body>
```



```
<h1>My Home Page</h1>
```

```
</body>
```

```
</html >
```

**16. Use the correct attribute to make sure the last paragraph gets the style as described in the style element.**

```
<!DOCTYPE html >
```

```
<html >
```

```
<head>
```

```
<style>
```

```
#special {
```

```
  color: gray;
```

```
  background-color: lightblue;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p>This is a paragraph. </p>
```

```
<p>This is a paragraph. </p>
```

```
<p 
```

```
</body>
```

```
</html >
```

**17. Use the HTML image attributes to set the image size to 250 pixels wide and 400 pixels high.**

```

```

**18. Specify alternate text for the image.**

The alternate text should say "Smiley".



Alternate text is useful when the image cannot be displayed, such as when the page is being read by a screen reader.

```

```

19. Clean this document with appropriate end tags.

```
<h1>This is a Heading   
<p>This is a paragraph. 
```

20. Use the correct HTML attribute and CSS to set the paragraph color to "blue".

```
<p =" ; ">This is a paragraph. </p>
```

21. Use CSS to align the center of the paragraph.

```
<p style=" : center; ">This is a paragraph. </p>
```

22. Use CSS to set the set the text size to 50 pixels.

```
<p style=" : 50px; ">This is a paragraph. </p>
```

23. Add an additional meaning to the word "degradation" in the paragraph below.

```
<p>  
WWF's mission is to stop the  degradation  of our  
planet's natural environment.  
</p>
```

24. Highlight the word "metropolitan" in the text below.

```
<h1>Tokyo</h1>
```





<p>

Tokyo is the capital of Japan, the most populous metropolitan  area in the world.

</p>

25. Highlight the word "FUN" in the text below.

<p>

HTML is  FUN  to learn!

</p>

26. Use an HTML element to add quotes around the letters "cool".

<p>

I am so  cool .

</p>

27. Letters „WHO “ in the text below are abbreviation of „World Health Organization “.

Use an HTML element to provide the specified abbreviation of "WHO".

<p>

The   "World Health Organization" > WHO</ >  
was founded in 1948.

</p>

28. Use the HTML comment tag to make a comment from the text "This is a comment".

<h1>This is a heading</h1>

This is a comment

<p>This is a paragraph. </p>

29. Use CSS to set the yellow background of the document (body).



```
<!DOCTYPE html >
<html >
<head>
<style e>


:yel low;

</styl e>
</head>
<body>

<h1>My Home Page</h1>

</body>
</html >
```

**30. Use CSS to set the text color of the document red.**

```
<!DOCTYPE html >
<html >
<head>
<style e>
body {: red; }
</styl e>
</head>
<body>

<h1>My Home Page</h1>

</body>
</html >
```

**31. Use CSS to make a yellow, 1 pixel thick frame around all paragraphs.**

```
<!DOCTYPE html >
<html >
<head>
<style e>
```



```

<div style="border: 1px solid black; width: 100px; height: 100px; display: inline-block;">
```

## CSS

### 32. Change the color of all <p> elements to "red".

Tip: Use the element selector.

```

<!DOCTYPE html >
<html >
<head>
<style >

</style >
</head >
<body >

<h1>This is a Heading</h1>
<p>This is a paragraph. </p>
<p>This is another paragraph. </p>

</body >
</html >
```

### 33. Change the color of the element with id = "para1" to "red".

Tip: Use the identification selector.



```
<!DOCTYPE html >
<html >
<head>
<style>

</style>
</head>
<body>

<h1>This is a Heading</h1>
<p id="para1">This is a paragraph. </p>
<p>This is another paragraph. </p>

</body>
</html >
```

#### 34. Add an external style sheet with a URL: "mystyle.css".

Tip: Add a <link> element to the <head>. The <link> element must have the "rel", "type" and "href" attributes.

```
<!DOCTYPE html >
<html >
<head>

</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph. </p>
<p>This is another paragraph. </p>

</body>
</html >
```



### 35. Remove all styles except the external sheet for "mystyle.css" style.

Tip: Remove the inner sheet and the embedding style.

```
<!DOCTYPE html >
<html >
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
<style>
p {
  color: red;
}
</style>
</head>
<body style="background-color: lightcyan">

<h1>This is a Heading</h1>
<p>This is a paragraph. </p>
<p>This is another paragraph. </p>

</body>
</html >
```

### 36. Set a "4px", "dotted" border for <p>.

Tip: Use the properties of the border and the width of the border.

```
<!DOCTYPE html >
<html >
<head>
<style>

</style>
</head>
<body>
```



```
<h1>This is a Heading</h1>
<p>This is a paragraph. </p>

</body>
</html >
```

### 37. Change the 3 border properties so that they show only the border at the top.

Tip: Use the border-**top** -style property, etc.

```
<!DOCTYPE html >
<html >
<head>
<style>
p {
border-style: dotted;
border-width: 4px;
border-color: red;
}
</style>
</head>
<body>
```

```
<h1>This is a Heading</h1>
<p>This is a paragraph. </p>

</body>
</html >
```

### 38. Set the left margin of <h1> to "20px".

Tip: Use the left field property.

```
<!DOCTYPE html >
<html >
<head>
```



```
<style>
h1 {
  background-color: lightblue;
}
</style>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph. </p>

</body>
</html >
```

**39. Set all pad for <p> to "50px".**

Tip: Use the pad property.

```
<!DOCTYPE html >
<html >
<head>
<style>
p {
  background-color: lightblue;
}
</style>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph. </p>

</body>
</html >
```



#### 40. Set the width <h1> to "50%".

Tip: Use the width property.

```
<!DOCTYPE html >
<html >
<head>
<style>
h1 {
  background-color: lightblue;
}
</style>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph. </p>

</body>
</html >
```

#### 41. Set div pad to "25px".

Tip: Use the pad property

```
<!DOCTYPE html >
<html >
<head>
<style>
div {
  background-color: lightblue;
  width: 200px;
}
</style>
</head>
<body>
```





```
<div>Lorem ipsum dolor sit amet, consectetur adipiscing elit,  
sed do eiusmod tempor incididunt ut labore et dolore magna  
aliqua. </div>
```

```
</body>  
</html>
```

## JavaScript

42. Shows the sum of **5 + 10** using two variables: **x** and **y**.

```
var  = ;  
var y = 10;  
document.getElementById("demo").innerHTML = x  y;
```

43. On a single line, declare three variables with the following names and values:

```
firstName = "John"  
lastName = "Doe"  
age = 35
```

```
var  = "John"  lastName =   =   
;
```

44. Notify the remainder when **15** is divided by **9**.

```
alert(15  9);
```

45. Use comments to describe the correct data type of the following variables:

```
var length = 16; //   
var lastName = "Johnson"; //   
var x = {
```



```
firstName: "John",  
lastName: "Doe"  
}; // 
```

46. Make the function return "Hello".

```
function myFunction() {  
   "";  
}  
document.getElementById("demo").innerHTML = myFunction();
```

47. Take the value „Volvo“ from the cars array.

```
var cars = ["Saab", "Volvo", "BMW"];  
var x = ;
```

48. Select the correct comparison operator to signal true true, when x is greater than y.

```
x = 10;  
y = 5;  
alert(x  y);
```

49. Select the correct comparison operator to signal true true, then x is NOT equal to y.

```
x = 10;  
y = 5;  
alert(x  y);
```

## jQuery

50. Use the appropriate selector to hide all <p> elements.

```
$ ( " " ). Hi de();
```



51. Use the jQuery method to hide the <p> element when you click on it.

```
$("#p").click(function() {  
    $(this).();  
});
```

52. Use the appropriate event to hide all <p> elements with "double-click".

```
$("#p"). (function() {  
    $(this).hide();  
});
```

53. If you press a keyboard key inside an <input> element, it must be hidden. Use the appropriate event to do it.

```
$("#input"). (function() {  
    $(this).hide();  
});
```

54. Use the jQuery method to hide the <p> element when you click on it.

```
$("#p").click(function() {  
    $(this).();  
});
```

55. Use the jQuery method to show the <p> element.

```
$("#button").click(function() {  
    $("#p").();  
});
```

56. Use the jQuery method to fade a <div> element.



Note: **The duration of the effect should be "slow".**

```
$("#div").();
```

**57. Use the jQuery method to fade a <div> element to an opacity of "0.2".**

```
$("#div").(, );
```

**58. Use the jQuery method to drag a <div> element up.**

```
$("#div").();
```

**59. Use the jQuery method to drag a <div> element down.**

```
$("#div").();
```

**60. Use the jQuery method to stop the animation effect of a <div> element.**

```
$("#div").();
```

**61. Use the jQuery method to return the text content of a <div> element.**

```
$("#div").();
```

**62. Use the jQuery method to return the URL of a link.**

```
$("#a").("href");
```

**63. Use the jQuery method to insert the text "YES!" at the end of an <p> element.**

```
$("#p").("YES! ");
```

**64. Use the jQuery method to remove only the child elements of a <div> element.**



```
$("#div").();
```

65. Use jQuery methods to set the height and width <div> of 500 pixels.

```
$("#div").()();
```

66. Use the jQuery method to get the height of the <div> element, except for the padding, border and field.

```
$("#div").();
```

67. Use the jQuery method to get the direct parent of the <span> element.

```
$("#span").();
```

## Bootstrap

68. Use contextual classes to add the following colors to lines:

Green first line green

second red

last line blue

```
<table class="table">
  <tr class="
```



```
</tr>
<tr class=" " >
<td>July</td>
<td>Dool ey</td>
<td>j ul y@exampl e. com</td>
</tr>
</tabl e>
```

**69. Add a Bootstrap class to group the buttons together.**

```
<di v class=" " >
<butt on class="btn btn- pri mary">Appl e</butt on>
<butt on class="btn btn- pri mary">Samsu ng</butt on>
<butt on class="btn btn- pri mary">Sony</butt on>
</di v>
```

**70. Use the correct class name to animate the progress bar.**

```
<di v class="progress">
<di v class="progress- bar progress- bar- stri ped "
rol e="progressbar"
styl e="wi dth: 70%">
</di v>
</di v>
```

**71. Add "divider" after "CSS" to the drop-down list.**

```
<di v class="dropdown">
<butt on data- toggl e="dropdown"
class="btn btn- pri mary dropdown- toggl e">
Dropdow n Exampl e
<span class="caret" ></span></butt on>
```



```
<ul class="dropdown-menu">
<li><a href="#">HTML</a></li>
<li><a href="#">CSS</a></li>
<li class=" " ></li>
<li><a href="#">JavaScript</a></li>
</ul>
</div>
```

## JAVA

72. Insert the missing part of the code below to display "Hello World".

```
public class MyClass {
    public static void main(String[] args) {
        . . . ("Hello World");
    }
}
```

73. Java comments are written in special characters. Insert the missing parts:

```
 This is a single-line comment
 This is a multi-line comment
```

74. Create a variable called z equal to x + y and display the result.

```
int x = 5;
int y = 10;
 = x + y;
System.out.println( );
```



75. Fill in the missing parts to create three variables of the same type using a comma-separated list:

```
 x = 5  y = 6  z = 50;  
System.out.println(x + y + z);
```

76. byte, short, int, long, float, double, boolean and char are called:

```
 data types.
```

77. Type conversion - convert the following double type (myDouble) to int type:

```
double myDouble = 9.78;  
int myInt =  myDouble;
```

78. Use the appropriate operator to increase the value of the variable x by 1.

```
int x = 10;  
 x;
```

79. Use the add assignment operator to add a value of 5 to the variable x.

```
int x = 10;  
x  5;
```

80. Use the appropriate method to connect two strings:

```
String firstName = "John ";  
String lastName = "Doe";  
System.out.println(firstName. (lastName));
```

81. Return the index (position) of the first occurrence of "e" in the following string:





```
String txt = "Hello Everybody";  
System.out.println(txt.());
```

**82. Insert the missing parts to complete the following switch action**

```
int day = 2;  
switch () {  
     1:  
        System.out.println("Saturday");  
        break;  
     2:  
        System.out.println("Sunday");  
    ;  
}
```

**83. Print i for as long, but keep in mind that i is less than 6.**

```
int i = 1;  
 (i < 6) {  
    System.out.println(i);  
    ;  
}
```

**84. Use the do/while outline to print i for as long, but keep in mind that i is less than 6.**

```
int i = 1;  
 {  
    System.out.println(i);  
    i++;  
}  
 (i < 6);
```



**85. Check the elements in the cars array.**

```
String[] cars = {"Volvo", "BMW", "Ford"};
 (String i : ) {
    System.out.println(i);
}
```

**86. Stop the cycle if i is 5.**

```
for (int i = 0; i < 10; i++) {
    if (i == 5) {
        ;
    }
    System.out.println(i);
}
```

**87. In the cycle, when the value is "4", go directly to the next value.**

```
for (int i = 0; i < 10; i++) {
    if (i == 4) {
        ;
    }
    System.out.println(i);
}
```

**88. Change the value from "Volvo" to "Opel" in the cars array.**

```
String[] cars = {"Volvo", "BMW", "Ford"};
 = ;
System.out.println(cars[0]);
```



89. Check the elements in the cars array.

```
String[] cars = {"Volvo", "BMW", "Ford"};
 (String i : ) {
    System.out.println(i);
}
```

90. Insert the missing parts to create a 2D array.

```
 myNumbers = { {1, 2, 3, 4}, {5, 6, 7} };
```

91. Insert the missing part to print the number 8 main using a specific myMethod keyword:

```
static int myMethod(int x) {
     5 + x;
}

public static void main(String[] args) {
    System.out.println(myMethod(3));
}
```

92. Create a class called MyClass.

```
public class 
```

93. Create a MyClass object called myObj.

```
  = new  ();
```

94. Fill in the missing parts to import the java.util.Scanner class from the Java API:

    ;

95. Print "Hello World" if x is greater than y.

```
int x = 50;
int y = 10;
 (x  y) {
    System.out.println("Hello World");
}
```

96. Insert the missing keyword to execute the code after try..catch, regardless of the result.

```
try {
    int[] myNumbers = {1, 2, 3};
    System.out.println(myNumbers[10]);
} catch (Exception e) {
    System.out.println("Something went wrong. ");
}  {
    System.out.println("The 'try catch' is finished. ");
}
```

97. Create and request a class constructor from MyClass

Follow the comments to insert the missing parts of the code below:

```
// Create a MyClass class
public class  {
    int ; // Create a class attribute x

    // Create a class constructor for the MyClass class
    public  () {
```



```
x = ; // Set the initial value for the class
attribute x to 5
}

public static void main(String[] args) {
// Create an myObj object of class MyClass (This will call
the constructor)
MyClass  = new MyClass();
// Print the value of x
System.out.println(.);
}
}
```

98. Insert the missing statement to get all the Customers columns in the table.

```
 * FROM Customers;
```

99. Write a statement that will select the City column from Customers in the table.

```
  Customers;
```

100. Select all the different values from the Country column in the Customers table.

```
  Country FROM Customers;
```

101. Select all entries, where the City column has value „Berlin“.

```
SELECT * FROM Customers
  = ;
```

102. Update the City column of all entries in the Customerstable.



```
 Customers  
 City = 'Oslo';
```

103. Use the correct function to return the number of entries to which Price is set to 18.

```
SELECT  (*)  
FROM Products  
 Price = 18;
```

104. Select all entries, where the first letter of City starts with something from "a" to "f".

```
SELECT * FROM Customers  
WHERE City LIKE '%';
```

105. Select all entries, where the first letter of City is NOT "a" or "c" or "f".

```
SELECT * FROM Customers  
WHERE City LIKE '%';
```

106. Select all entries, where the first letter of City "e" or "c" or "s".

```
SELECT * FROM Customers  
WHERE City LIKE '%';
```

107. Select all entries, where the second letter of City "a".

```
SELECT * FROM Customers  
WHERE City LIKE '%';
```

108. Update the values of City and of Country.



```
 Customers  
 City = 'Oslo'   
 = 'Norway'  
WHERE CustomerID = 32;
```



## TEST QUESTIONS

### Question 1. What does HTML mean?

1. Hyper Tag Markup Language
2. Hyper Text Markup Language
3. Hyperlinks Text Mark Language
4. Hyperlinking Text Marking Language

### Question 2. What symbol indicates a tag?

1. Angle brackets e.g. <>
2. Curved brackets, e.g. {,}
3. Quotes ','
4. Exclamation mark

### Question 3. Which of the following is a primary keyword?

1. Header
2. Bold
3. Body
4. Image

### Question 4. A CSS file can only be attached to one HTML file.

1. True
2. False

### Question 5. What is the correct break tag?

1. "brk /"
2. "line /"
3. "bk /"
4. "br /"

### Question 6. What does CSS mean?

1. Computing Style Sheet
2. Creative Style System
3. Cascading Style Sheet
4. Creative Styling Sheet





**Question 7. Where should the CSS file be specified in the HTML file?**

1. Before each HTML code
2. At the end of the HTML code
3. In the head section
4. In body section

**Question 8. What is the correct format for aligning written content to the center of the page in CSS?**

1. Text-align:center;
2. Font-align:center;
3. Text:align-center;
4. Font:align-center;

**Question 9. What is the correct format for changing the div background color in CSS?**

1. Bg-color:red;
2. Background:red;
3. Background-colour:red;
4. Background-color:red;

**Question 10. What is the correct format for div?**

1. Div-id=example
2. Div id="example"
3. Div="example"
4. Div.example

**Question 11. This is a document or information resource that is suitable for the World Wide Web and can be accessed through a web browser and displayed on a computer screen.**

1. Web browser
2. Web page
3. Html

**Question 12. How can you open a link in a new browser window?**

1. href = "url" new>
2. href = "url" target = "\_ blank">
3. href = "url" target = "new">



**Question 13. The network is based on:**

1. Photos
2. Text
3. Information
4. HTML

**Question 14. What is a web browser?**

1. Something in the taskbar
2. It is used to create web pages
3. Software application for retrieving and presenting information on the Internet

**Question 15. What does FTP mean?**

1. Files To Put online
2. File Transfer Protocol
3. File Transfer Please

**Question 16. What is a web hosting?**

1. Domain name
2. What people see with a browser
3. On-line space for website and data

**Question 17. The correct HTML tag to make the text italic is "italic".**

1. True
2. False

**Question 18. What is the correct HTML code to insert an image?**

1. `Img href="image.gif />`
2. `Image src="image.gif" />`
3. `Img src="image.gif" />`

**Question 19. CSS does its job by stepping between \_\_\_\_\_ and \_\_\_\_\_**

1. TEXT, IMAGES
2. HTML, BROWSER
3. HPPT, VISITOR

**Question 20. What is the correct HTML code to create a hyperlink?**



1. >http://www.vbimedia.com
2. url="http://www.vbimedia.com">VBIMedia.com
3. href="http://www.vbimedia.com">VBIMEDIA

**Question 21. Choose the correct HTML tag for the largest title**

1. H6
2. H3
3. H1

**Question 22. HTML provides 5 header tags h1 to h5**

1. True
2. False

**Question 23. What is the correct HTML code to add a background color?**

1. Background>yellow
2. Body bgcolor="#000000"
3. Body background< #227645>

**Question 24. The domain name is an identification tag.**

1. True
2. False

**Question 25. CSS avoids all \_ \_ \_ \_ formatting issues**

1. TEXT
2. HTTP
3. HTML

**Question 26. What is the predominant language for creating web pages?**

1. HTML
2. PHP
3. CSS

**Question 27. What is a web server?**

1. A computer program that delivers content
2. World Wide Web
3. Remote computer used for network computers



**Question 28. An HTML element known as <ins> defines**

1. Instantiation
2. Insertion
3. Incapsulation
4. Instance

**Question 29. External CSS usually sets the style for?**

1. Multiple elements
2. Single HTML page
3. Multiple HTML pages
4. Single element

**Question 30. The HTML <div> element describes:**

1. Division
2. Divided
3. Divisor
4. Diversion

**Question 31. URL encoding replaces a space in a URL with a character**

1. %
2. ?
3. &
4. +

**Question 32. The null data type is a / an**

1. Object
2. Indefinite
3. Number
4. Function

**Question 33. The series of characters forms .....**

1. Syntax analysis
2. Template expression
3. Regular expression
4. Detailed expression

**Question 34. In HTML5, it is known that the default character set is from**



1. UTF-6
2. UTF-7
3. UTF-8
4. UTF-5

**Question 35. How to insert a comment in a CSS file?**

1. `' this is a comment`
2. `// this is a comment //`
3. `// this is a comment`
4. `/* this is a comment */`



## ANSWERS TO PRACTICAL EXERCISES AND TESTS

### Answers to exercises

#### HTML

1. Add a "hint" to the paragraph below with the text "All About W3Schools".

```
<p title="About W3Schools">W3Schools is a web developer's site. </p>
```

2. Set the image size to 250 pixels wide and 400 pixels high.

```

```

3. Use the correct HTML tag to add a title with the text „London“.

```
<h1>London</h1>
```

```
<p>London is the capital city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants. </p>
```

4. Add a horizontal rule between the title and paragraph.

```
<h1>London</h1>
```

```
<hr>
```

```
<p>London is the capital city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants. </p>
```

5. Use the correct HTML tag to add a paragraph with the text "Hello World!".



```
<html >  
<body>  
<p>Hello W  
</body>  
</html >
```

6. Clean this document with appropriate end tags.

```
<h1>This is a Heading </h1>  
<p>This is a paragraph. </p>
```

7. Use the correct HTML attribute and CSS to set the paragraph color to "blue".

```
<p style="color:blue;">This is a paragraph. </p>
```

8. Use CSS to set the paragraph font to "courier".

```
<p style="font-family: courier;">This is a paragraph. </p>
```

9. Use CSS to align the center of the paragraph.

```
<p style="text-align: center;">This is a paragraph. </p>
```

10. Add an additional meaning to the word "degradation" in the paragraph below.

```
<p>  
WWF's mission is to stop the <strong>degradation</strong> of our  
planet's natural environment.  
</p>
```

11. Highlight the word "FUN" in the text below.



```
<p>  
HTML is  FUN  to learn!  
</p>
```

12. Add a line by crossing out the letters "blue" in the text below.

```
<p>  
My favorite color is  blue  red.  
</p>
```

13. Use an HTML element to add quotes around the letters "cool".

```
<p>  
I am so  cool .  
</p>
```

14. Make the text below go from right to left.

```
What a beautiful day!
```

15. Use CSS to set the document font to "courier".

```
<!DOCTYPE html >  
<html >  
<head>  
<style >  
body {  : couri er; }  
</style >  
</head >  
<body >  
  
<h1>My Home Page</h1 >
```





```
</body>  
</html >
```

**16. Use the correct attribute to make sure the last paragraph gets the style as described in the style element.**

```
<!DOCTYPE html >  
<html >  
<head>  
<style>  
#special {  
  color: gray;  
  background-color: lightblue;  
}  
</style>  
</head>  
<body>  
  
<p>This is a paragraph. </p>  
<p>This is a paragraph. </p>  
<p id="special">This is a paragraph. </p>  
  
</body>  
</html >
```

**17. Use the HTML image attributes to set the image size to 250 pixels wide and 400 pixels high.**

```

```

**18. Specify alternate text for the image.**

The alternate text should say "Smiley".



Alternate text is useful when the image cannot be displayed, such as when the page is being read by a screen reader.

```

```

19. Clean this document with appropriate end tags.

```
<h1>This is a Heading </h1>
```

```
<p>This is a paragraph. </p>
```

20. Use the correct HTML attribute and CSS to set the paragraph color to "blue".

```
<p style="color:blue;">This is a paragraph. </p>
```

21. Use CSS to align the center of the paragraph.

```
<p style="text-align:center;">This is a paragraph. </p>
```

22. Use CSS to set the set the text size to 50 pixels.

```
<p style="font-size:50px;">This is a paragraph. </p>
```

23. Add an additional meaning to the word "degradation" in the paragraph below.

```
<p>  
WWF's mission is to stop the <strong>degradation</strong> of our  
planet's natural environment.  
</p>
```

24. Highlight the word "metropolitan" in the text below.

```
<h1>Tokyo</h1>
```



<p>

Tokyo is the capital of Japan, the most populous metropolitan  area in the world.

</p>

25. Highlight the word "FUN" in the text below.

<p>

HTML is  FUN  to learn!

</p>

26. Use an HTML element to add quotes around the letters "cool".

<p>

I am so  cool .

</p>

27. Letters „WHO “ in the text below are abbreviation of „World Health Organization “.

Use an HTML element to provide the specified abbreviation of "WHO".

<p>

The   WHO  was founded in 1948.

</p>

28. Use the HTML comment tag to make a comment from the text "This is a comment".

<h1>This is a heading</h1>

This is a comment

<p>This is a paragraph. </p>

29. Use CSS to set the yellow background of the document (body).



```
<!DOCTYPE html >
<html >
<head>
<style e>
body {
background: yellow;
}
</style e>
</head>
<body>

<h1>My Home Page</h1>

</body>
</html >
```

**30. Use CSS to set the text color of the document red.**

```
<!DOCTYPE html >
<html >
<head>
<style e>
body { color : red; }
</style e>
</head>
<body>

<h1>My Home Page</h1>

</body>
</html >
```

**31. Use CSS to make a yellow, 1 pixel thick frame around all paragraphs.**

```
<!DOCTYPE html >
<html >
<head>
<style e>
```



```
p { border : 1px solid yellow ; }
</style>
</head>
<body>

<p>This is a paragraph. </p>
<p>This is a paragraph. </p>
<p>This is a paragraph. </p>

</body>
</html >
```

## CSS

**32. Change the color of all <p> elements to "red".**

```
<!DOCTYPE html >
<html >
<head>
<style>
p {
  color: red;
}
</style>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph. </p>
<p>This is another paragraph. </p>

</body>
</html >
```

**33. Change the color of the element with id = "para1" to "red".**



```
<!DOCTYPE html >
<html >
<head>
<style>
#para1 {
  color: red;
}
</style>
</head>
<body>

<h1>This is a Heading</h1>
<p id="para1">This is a paragraph. </p>
<p>This is another paragraph. </p>

</body>
</html >
```

**34. Add an external style sheet with a URL: "mystyle.css".**

```
<!DOCTYPE html >
<html >
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph. </p>
<p>This is another paragraph. </p>

</body>
</html >
```



**35. Remove all styles except the external sheet for "mystyle.css" style.**

```
<!DOCTYPE html >
<html >
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph. </p>
<p>This is another paragraph. </p>

</body>
</html >
```

**36. Set a "4px", "dotted" border for <p>.**

```
<!DOCTYPE html >
<html >
<head>
<style >
p {
border-style: dotted;
border-width: 4px;
}
</style >
</head >
<body >

<h1>This is a Heading</h1>
<p>This is a paragraph. </p>

</body >
</html >
```



**37. Change the 3 border properties so that they show only the border at the top.**

```
<!DOCTYPE html >
<html >
<head>
<style>
p {
border-top-style: dotted;
border-top-width: 4px;
border-top-color: red;
}
</style>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph. </p>

</body>
</html >
```

**38. Set the left margin of <h1> to "20px".**

```
<!DOCTYPE html >
<html >
<head>
<style>
h1 {
background-color: lightblue;
margin-left: 20px;
}
</style>
</head>
<body>
```





```
<h1>This is a Heading</h1>  
<p>This is a paragraph. </p>  
  
</body>  
</html >
```

**39. Set all pad for <p> to "50px".**

```
<!DOCTYPE html >  
<html >  
<head>  
<style>  
p {  
  background-color: lightblue;  
  padding: 50px;  
}  
</style>  
</head>  
<body>  
  
<h1>This is a Heading</h1>  
<p>This is a paragraph. </p>  
  
</body>  
</html >
```

**40. Set the width <h1> to "50%".**

```
<!DOCTYPE html >  
<html >  
<head>  
<style>  
h1 {  
  background-color: lightblue;
```



```
width: 50%;  
}  
</style>  
</head>  
<body>  
  
<h1>This is a Heading</h1>  
<p>This is a paragraph.</p>  
  
</body>  
</html >
```

#### 41. Set div pad to "25px".

```
<!DOCTYPE html >  
<html >  
<head>  
<style>  
div {  
  background-color: lightblue;  
  width: 200px;  
  padding: 25px;  
}  
</style>  
</head>  
<body>  
  
<div>Lorem ipsum dolor sit amet, consectetur adipiscing elit,  
sed do eiusmod tempor incididunt ut labore et dolore magna  
aliqua. </div>  
  
</body>  
</html >
```



## JavaScript

42. Shows the sum of `5 + 10` using two variables: `x` and `y`.

```
var x = 5;  
var y = 10;  
document.getElementById("demo").innerHTML = x + y;
```

43. On a single line, declare three variables with the following names and values:

```
firstName = "John"  
lastName = "Doe"  
age = 35
```

```
var firstName = "John", lastName = "Doe", age = 35;  
;
```

44. Notify the remainder when `15` is divided by `9`.

```
alert(15 % 9);
```

45. Use comments to describe the correct data type of the following variables:

```
var length = 16; // Number  
var lastName = "Johnson"; // String  
var x = {  
  firstName: "John",  
  lastName: "Doe"  
}; // Object
```

46. Make the function return "Hello".

```
function myFunction() {  
  return "Hello";  
}
```



```
}  
document.getElementById("demo").innerHTML = myFunction();
```

47. Take the value „Volvo“ from the cars array.

```
var cars = ["Saab", "Volvo", "BMW"];  
var x = cars[1];
```

48. Select the correct comparison operator to signal true, when x is greater than y.

```
x = 10;  
y = 5;  
alert(x > y);
```

49. Select the correct comparison operator to signal true, then x is NOT equal to y.

```
x = 10;  
y = 5;  
alert(x != y);
```

## jQuery

50. Use the appropriate selector to hide all <p> elements.

```
$( "p" ).hide();
```

51. Use the jQuery method to hide the <p> element when you click on it.

```
$( "p" ).click(function() {  
$(this).hide();  
});
```

52. Use the appropriate event to hide all <p> elements with "double-click".



```
$("#p"). (function() {  
    $(this).hide();  
});
```

**53. If you press a keyboard key inside an <input> element, it must be hidden. Use the appropriate event to do it.**

```
$("#input"). (function() {  
    $(this).hide();  
});
```

**54. Use the jQuery method to hide the <p> element when you click on it.**

```
$("#p").click(function() {  
    $(this). ();  
});
```

**55. Use the jQuery method to show the <p> element.**

```
$("#button").click(function() {  
    $("#p"). ();  
});
```

**56. Use the jQuery method to fade a <div> element.**

```
$("#div"). ( );
```

**57. Use the jQuery method to fade a <div> element to an opacity of "0.2".**

```
$("#div"). (,  );
```

**58. Use the jQuery method to drag a <div> element up.**



```
$("#div").slideUp();
```

**59. Use the jQuery method to drag a <div> element down.**

```
$("#div").slideDown();
```

**60. Use the jQuery method to stop the animation effect of a <div> element.**

```
$("#div").stop();
```

**61. Use the jQuery method to return the text content of a <div> element.**

```
$("#div").text();
```

**62. Use the jQuery method to return the URL of a link.**

```
$("#a").attr("href");
```

**63. Use the jQuery method to insert the text "YES!" at the end of an <p> element.**

```
$("#p").append("YES!");
```

**64. Use the jQuery method to remove only the child elements of a <div> element.**

```
$("#div").empty();
```

**65. Use jQuery methods to set the height and width <div> of 500 pixels.**

```
$("#div").height(500).width(500);
```

**66. Use the jQuery method to get the height of the <div> element, except for the padding, border and field.**



```
$("#div").height();
```

67. Use the jQuery method to get the direct parent of the <span> element.

```
$("#span").parent();
```

## Bootstrap

68. Use contextual classes to add the following colors to lines:

Green first line green

second red

last line blue

```
<table class="table">
  <tr class="success">
    <td>John</td>
    <td>Doe</td>
    <td>john@example.com</td>
  <tr>
    <tr class="danger">
      <td>Mary</td>
      <td>Moe</td>
      <td>mary@example.com</td>
    </tr>
    <tr class="info">
      <td>July</td>
      <td>Dool ey</td>
      <td>july@example.com</td>
    </tr>
  </table>
```



**69. Add a Bootstrap class to group the buttons together.**

```
<div class="btn-group" >
  <button class="btn btn-primary">Apply</button>
  <button class="btn btn-primary">Samsung</button>
  <button class="btn btn-primary">Sony</button>
</div>
```

**70. Use the correct class name to animate the progress bar.**

```
<div class="progress">
  <div class="progress-bar progress-bar-striped active"
  role="progressbar"
  style="width: 70%" >
</div>
</div>
```

**71. Add "divider" after "CSS" to the drop-down list.**

```
<div class="dropdown">
  <button data-toggle="dropdown"
  class="btn btn-primary dropdown-toggle">
  Dropdown Example
  <span class="caret"></span></button>
  <ul class="dropdown-menu">
  <li><a href="#">HTML</a></li>
  <li><a href="#">CSS</a></li>
  <li class="divider"></li>
  <li><a href="#">JavaScript</a></li>
  </ul>
</div>
```





## JAVA

72. Insert the missing part of the code below to display "Hello World".

```
public class MyClass {  
    public static void main(String[] args) {  
        System.out.println("Hello World");  
    }  
}
```

73. Java comments are written in special characters. Insert the missing parts:

```
// This is a single-line comment  
/* This is a multi-line comment */
```

74. Create a variable called z equal to x + y and display the result.

```
int x = 5;  
int y = 10;  
int z = x + y;  
System.out.println(z);
```

75. Fill in the missing parts to create three variables of the same type using a comma-separated list:

```
int x = 5, y = 6, z = 50;  
System.out.println(x + y + z);
```

76. byte, short, int, long, float, double, boolean and char are called:

```
primitive data types.
```



77. Type conversion - convert the following double type (myDouble) to int type:

```
double myDouble = 9.78;  
int myInt =  myDouble;
```

78. Use the appropriate operator to increase the value of the variable x by 1.

```
int x = 10;  
 x;
```

79. Use the add assignment operator to add a value of 5 to the variable x.

```
int x = 10;  
x  5;
```

80. Use the appropriate method to connect two strings:

```
String firstName = "John ";  
String lastName = "Doe";  
System.out.println(firstName. (lastName));
```

81. Return the index (position) of the first occurrence of "e" in the following string:

```
String txt = "Hello Everybody";  
System.out.println(txt. ());
```

82. Insert the missing parts to complete the following switch action

```
int day = 2;  
switch () {  
     1:  
        System.out.println("Saturday");
```



```
break;
case 2:
System.out.println("Sunday");
break;
}
```

**83. Print i for as long, but keep in mind that i is less than 6.**

```
int i = 1;
while (i < 6) {
System.out.println(i);
i++;
}
```

**84. Use the do/while outline to print i for as long, but keep in mind that i is less than 6.**

```
int i = 1;
do {
System.out.println(i);
i++;
}
while (i < 6);
```

**85. Check the elements in the cars array.**

```
String[] cars = {"Volvo", "BMW", "Ford"};
for (String i : cars) {
System.out.println(i);
}
```

**86. Stop the cycle if i is 5.**



```
for (int i = 0; i < 10; i++) {  
    if (i == 5) {  
        break;  
    }  
    System.out.println(i);  
}
```

**87. In the cycle, when the value is "4", go directly to the next value.**

```
for (int i = 0; i < 10; i++) {  
    if (i == 4) {  
        continue;  
    }  
    System.out.println(i);  
}
```

**88. Change the value from "Volvo" to "Opel" in the cars array.**

```
String[] cars = {"Volvo", "BMW", "Ford"};  
cars[0] = "Opel";  
System.out.println(cars[0]);
```

**89. Check the elements in the cars array.**

```
String[] cars = {"Volvo", "BMW", "Ford"};  
for (String i : cars) {  
    System.out.println(i);  
}
```

**90. Insert the missing parts to create a 2D array.**



```
int[] myNumbers = { {1, 2, 3, 4}, {5, 6, 7} };
```

91. Insert the missing part to print the number 8 main using a specific myMethod keyword:

```
static int myMethod(int x) {  
    return 5 + x;  
}  
  
public static void main(String[] args) {  
    System.out.println(myMethod(3));  
}
```

92. Create a class called MyClass.

```
public class MyClass
```

93. Create a MyClass object called myObj.

```
MyClass myObj = new MyClass ();
```

94. Fill in the missing parts to import the java.util.Scanner class from the Java API:

```
import java.util.Scanner;
```

95. Print "Hello World" if x is greater than y.

```
int x = 50;  
int y = 10;  
if (x > y) {  
    System.out.println("Hello World");  
}
```



96. Insert the missing keyword to execute the code after try..catch, regardless of the result.

```
try {
    int[] myNumbers = {1, 2, 3};
    System.out.println(myNumbers[10]);
} catch (Exception e) {
    System.out.println("Something went wrong. ");
}  {
    System.out.println("The 'try catch' is finished. ");
}
```

97. Create and request a class constructor from MyClass

```
// Create a MyClass class
public class  {
    int ; // Create a class attribute x

    // Create a class constructor for the MyClass class
    public  () {
        x = ; // Set the initial value for the class
        attribute x to 5
    }

    public static void main(String[] args) {
        // Create an myObj object of class MyClass (This will call
        the constructor)
        MyClass  = new MyClass();
        // Print the value of x
        System.out.println(.);
    }
}
```



```
}
```

98. Insert the missing statement to get all the Customers columns in the table.

```
SELECT * FROM Customers;
```

99. Write a statement that will select the City column from Customers in the table.

```
SELECT City FROM Customers;
```

100. Select all the different values from the Country column in the Customers table.

```
SELECT DISTINCT Country FROM Customers;
```

101. Select all entries, where the City column has value „Berlin“.

```
SELECT * FROM Customers  
WHERE City = 'Berlin';
```

102. Update the City column of all entries in the Customerstable.

```
UPDATE Customers  
SET City = '0slo';
```

103. Use the correct function to return the number of entries to which Price is set to 18.

```
SELECT COUNT (*)  
FROM Products  
WHERE Price = 18;
```



104. Select all entries, where the first letter of City starts with something from "a" to "f".

```
SELECT * FROM Customers  
WHERE City LIKE '[a-f]%' ;
```

105. Select all entries, where the first letter of City is NOT "a" or "c" or "f".

```
SELECT * FROM Customers  
WHERE City LIKE '[^acf]%' ;
```

106. Select all entries, where the first letter of City "e" or "c" or "s".

```
SELECT * FROM Customers  
WHERE City LIKE '[ecs]%' ;
```

107. Select all entries, where the second letter of City "a".

```
SELECT * FROM Customers  
WHERE City LIKE ' _a%' ;
```

108. Update the values of City and of Country.

```
UPDATE Customers  
SET City = 'Oslo' ,  
Country = 'Norway'  
WHERE CustomerID = 32;
```





## Answers to test questions

**Question 1. What does HTML mean?**

2. Hyper Text Markup Language

**Question 2. What symbol indicates a tag?**

1. Angle brackets e.g. <,>

**Question 3. Which of the following is a primary keyword?**

3. Body

**Question 4. A CSS file can only be attached to one HTML file.**

2. False

**Question 5. What is the correct break tag?**

4. "br /"

**Question 6. What does CSS mean?**

3. Cascading Style Sheet

**Question 7. Where should the CSS file be specified in the HTML file?**

3. In the head section



**Question 8. What is the correct format for aligning written content to the center of the page in CSS?**

1. Text-align:center;

**Question 9. What is the correct format for changing the div background color in CSS?**

4. Background-color:red;

**Question 10. What is the correct format for div?**

2. Div id="example"

**Question 11. This is a document or information resource that is appropriate for World Wide Web and can be accessed through a web browser and displayed on computer screen.**

2. Web page

**Question 12. How can you open a link in a new browser window?**

2. href = "url" target = "\_ blank">

**Question 13. The network is based on:**

4. HTML

**Question 14. What is a web browser?**



3. Software application for retrieving and presenting information on the Internet

**Question 15. What does FTP mean?**

2. File Transfer Protocol

**Question 16. What is a web hosting?**

3. On-line space for website and data

**Question 17. The correct HTML tag to make the text italic is "italic".**

2. False

**Question 18. What is the correct HTML code to insert an image?**

3. `Img src="image.gif" />`

**Question 19. CSS does its job by stepping between \_\_\_\_\_ and \_\_\_\_\_**

2. HTML, BROWSER

**Question 20. What is the correct HTML code to create a hyperlink?**

3. `href="http://www.vbimedia.com">VBIMEDIA`

**Question 21. Choose the correct HTML tag for the largest title**

3. H1



**Question 22. HTML provides 5 header tags h1 to h5**

2. False

**Question 23. What is the correct HTML code to add a background color?**

2. Body bgcolor="#000000"

**Question 24. The domain name is an identification tag.**

1. True

**Question 25. CSS avoids all \_ \_ \_ \_ formatting issues**

3. HTML

**Question 26. What is the predominant language for creating web pages?**

1. HTML

**Question 27. What is a web server?**

1. A computer program that delivers content

**Question 28. An HTML element known as <ins> defines**

2. Insertion



**Question 29. External CSS usually sets the style for?**

3. Multiple HTML pages

**Question 30. The HTML <div> element describes:**

1. Division

**Question 31. URL encoding replaces a space in a URL with a character**

4. +

**Question 32. The null data type is a / an**

1. Object

**Question 33. The series of characters forms .....**

3. Regular expression

**Question 34. In HTML5, it is known that the default character set is from**

3. UTF-8

**Question 35. How to insert a comment in a CSS file?**

4. /\* this is a comment \*/