W2BUSINESS
QA Academy
Wroclaw - Spring 2018

Chapter “Software Development basics”

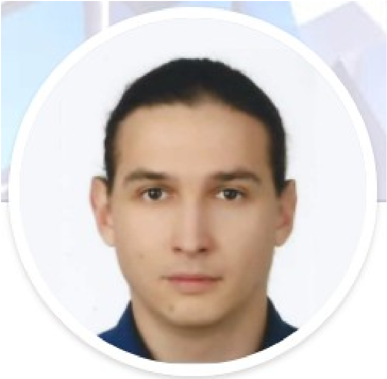
Lectors

Svitlana Samko

https://www.linkedin.com/in/svitlana-samko-b87532114/

Senior Developer in Test
over 10 years web development practice
over 10 delivered software projects for middle and large business

My most beneficial skill: *I like to learn business from the inside. Only so one can be sure that we build right product in the right way at any stage of development process.*



Andrii Stepura

https://www.linkedin.com/in/andriistepura/

Senior Quality Assurance Automation Engineer
over 14 years web development practice
over 300 delivered web projects as PO / Dev / Analyst / QA

My most beneficial skill: *Imagination to think like a stakeholder. Every piece of software starts from an idea. The first written code lines are just a half of the delivery of that idea.*

Definition of done

1 **Introduction to IT:**

- Introduction to IT in basic terms
- Software theory
  - SW goals, SW types, benefits
  - Software development life cycle, models
- Fundamentals of Testing

2 **Software Development basics:**

- Data
  - Formats - Text, graphic, binary
- Languages
  - Formal XML, HTML, CSS, JSON
  - Programming Javascript, PHP
- Test Levels and Types
- Static Techniques
- Review

Software Development basics

What do you know about...

Data ?

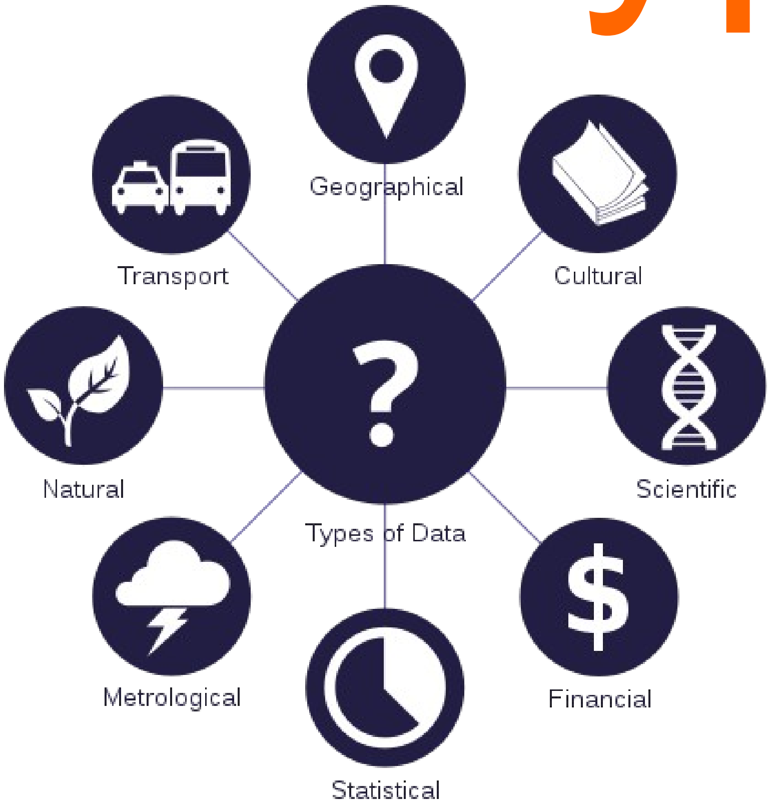
Data

(or datum – a single unit of data) requires interpretation to become information. To translate data to information, there must be several known factors considered. The factors involved are determined by the creator of the data and the desired information.

!

© https://en.wikipedia.org/wiki/Data_(computing)

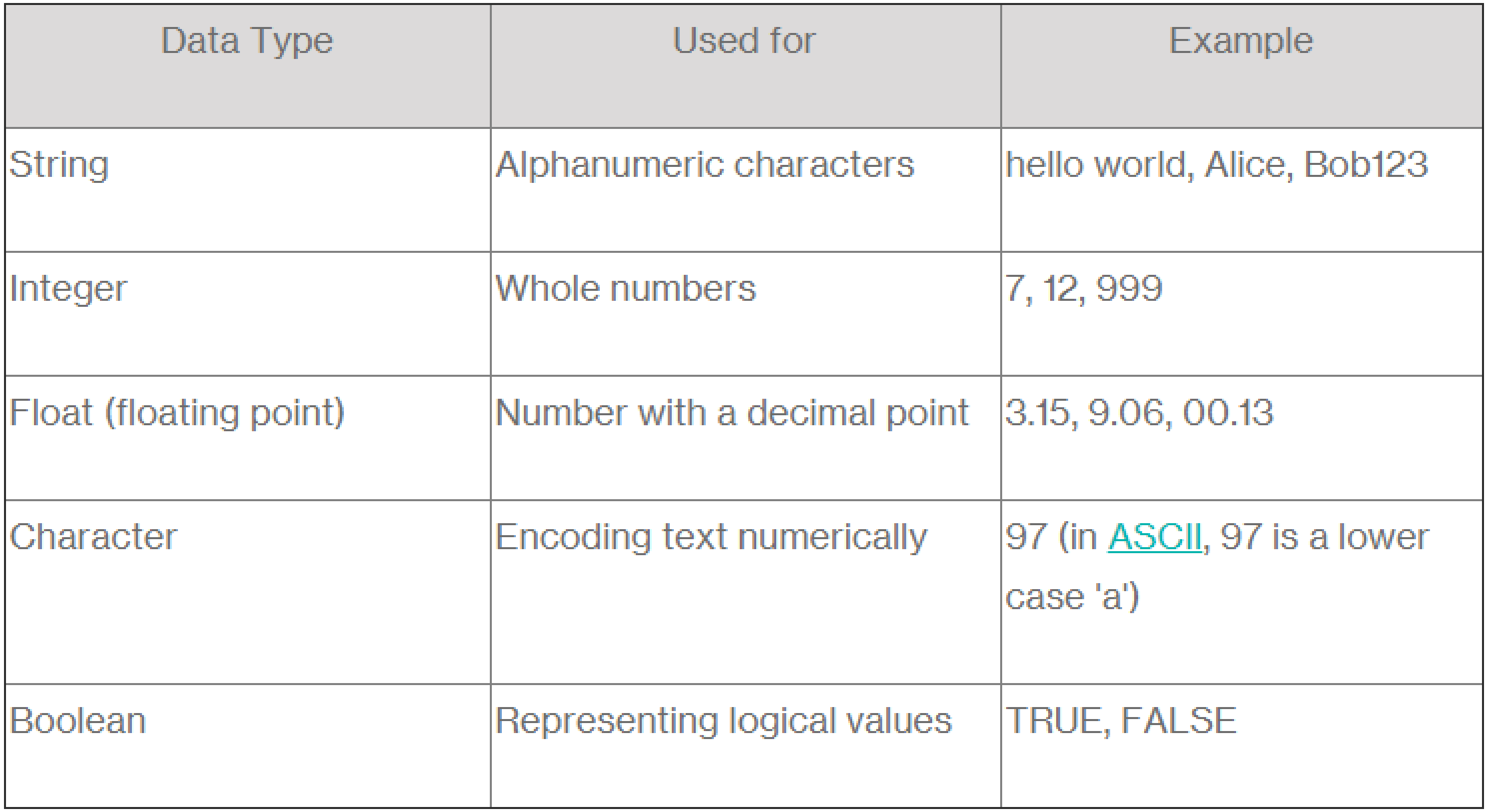
Data types



!

© https://en.wikipedia.org/wiki/Data_(computing)

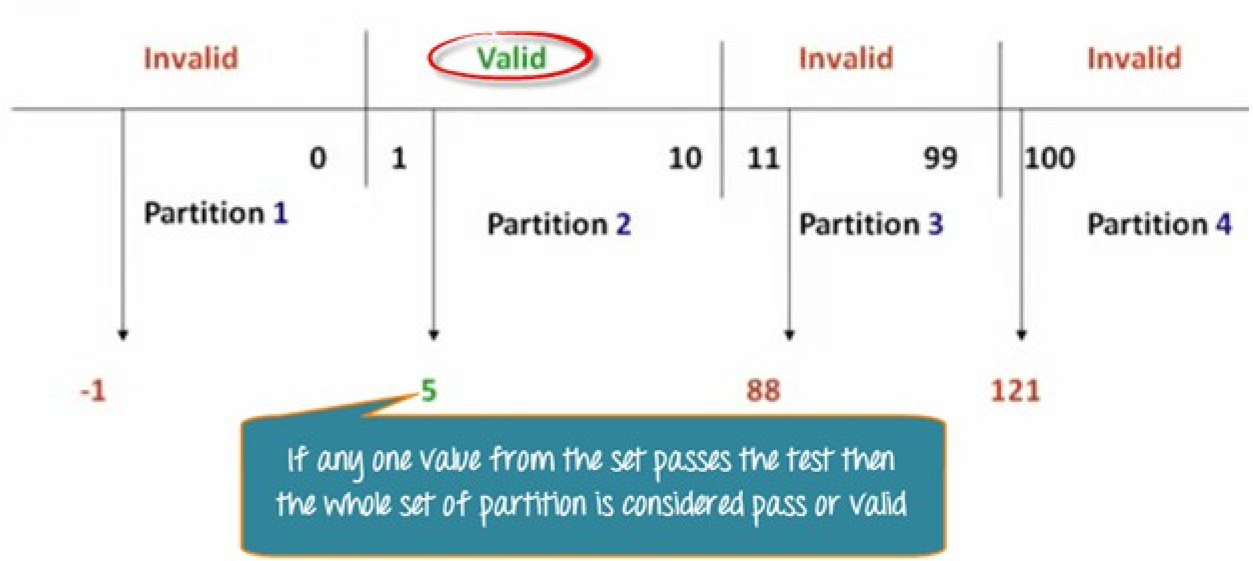
Data types



Data Type	Used for	Example
String	Alphanumeric characters	hello world, Alice, Bob123
Integer	Whole numbers	7, 12, 999
Float (floating point)	Number with a decimal point	3.15, 9.06, 00.13
Character	Encoding text numerically	97 (in ASCII, 97 is a lower case 'a')
Boolean	Representing logical values	TRUE, FALSE

© http://searchmicroservices.techtarget.com/definition/data-type

ISTQB equivalence partitioning examples:

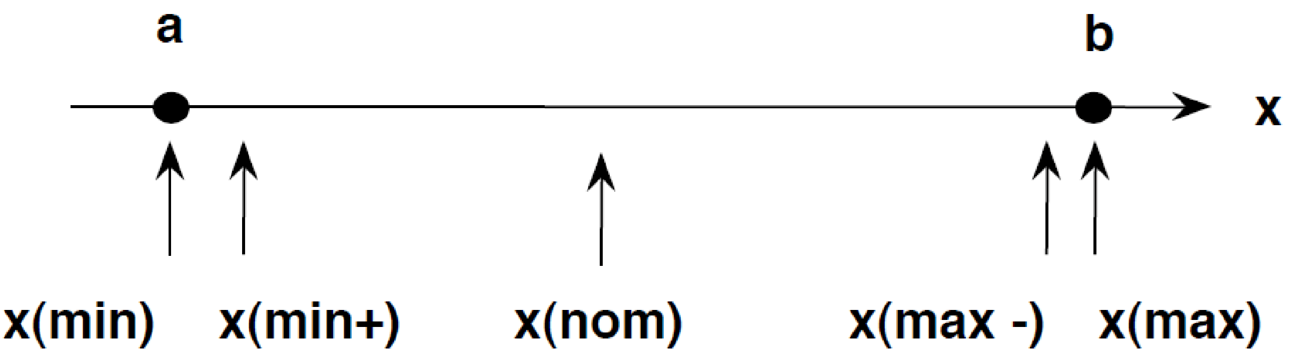


If any one value from the set passes the test then the whole set of partition is considered pass or valid

© https://www.guru99.com/equivalence-partitioning-boundary-value-analysis.html

Software Development basics

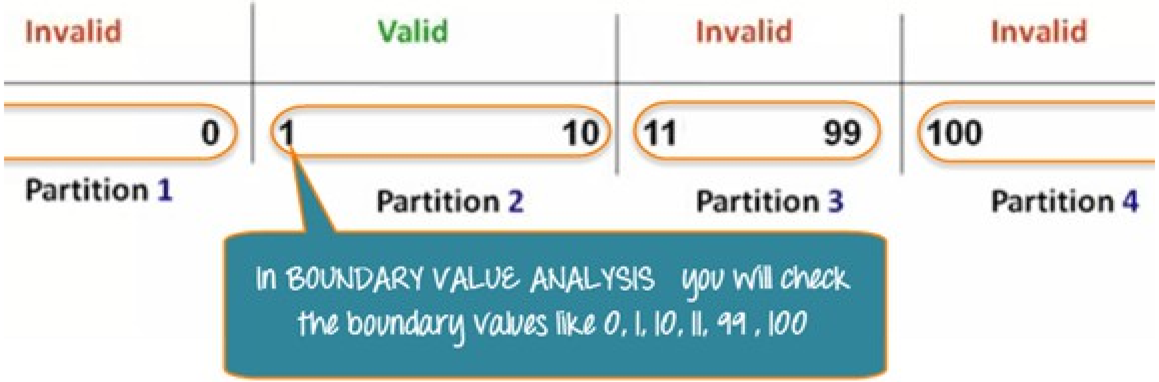
ISTQB Boundary Value Analysis:



!

© https://www.guru99.com/equivalence-partitioning-boundary-value-analysis.html

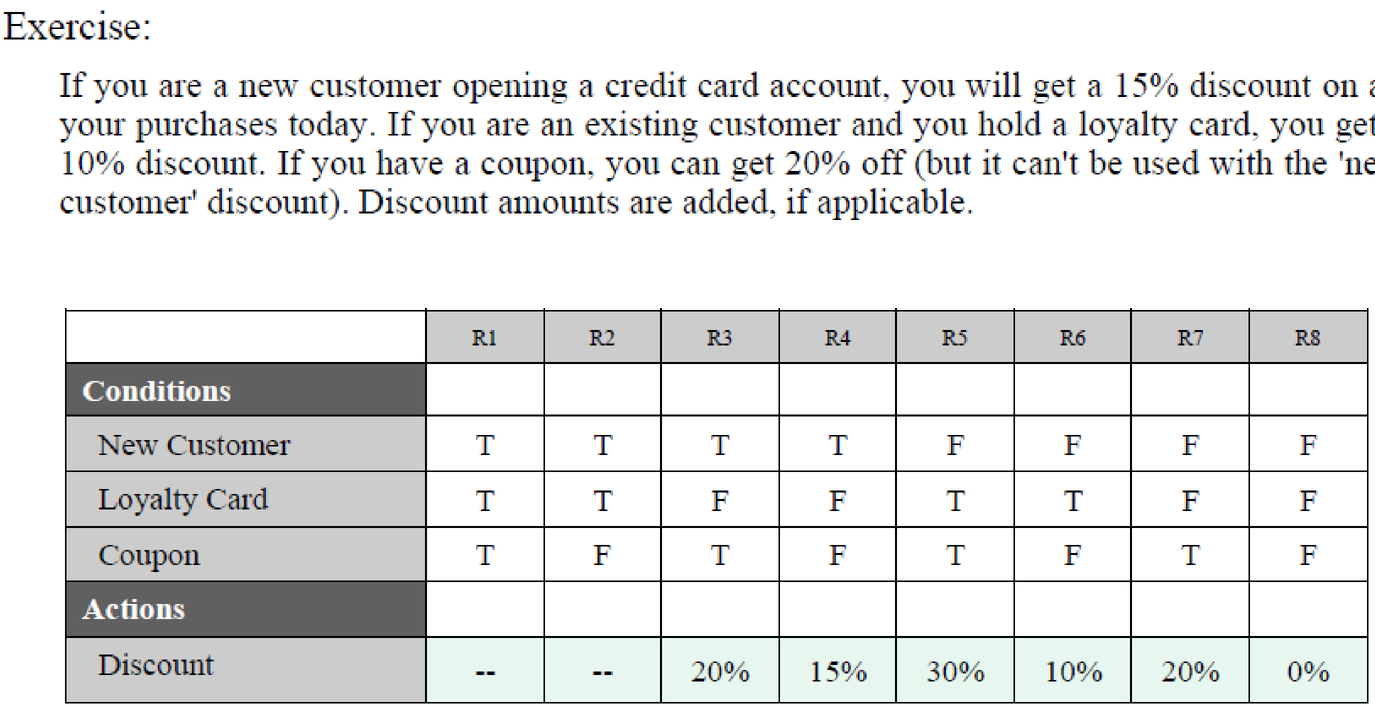
ISTQB Boundary Value Analysis & Equivalence Partitioning:



In BOUNDARY VALUE ANALYSIS you will check the boundary values like 0, 1, 10, 11, 99, 100

© https://www.guru99.com/equivalence-partitioning-boundary-value-analysis.html

ISTQB Decision Table:



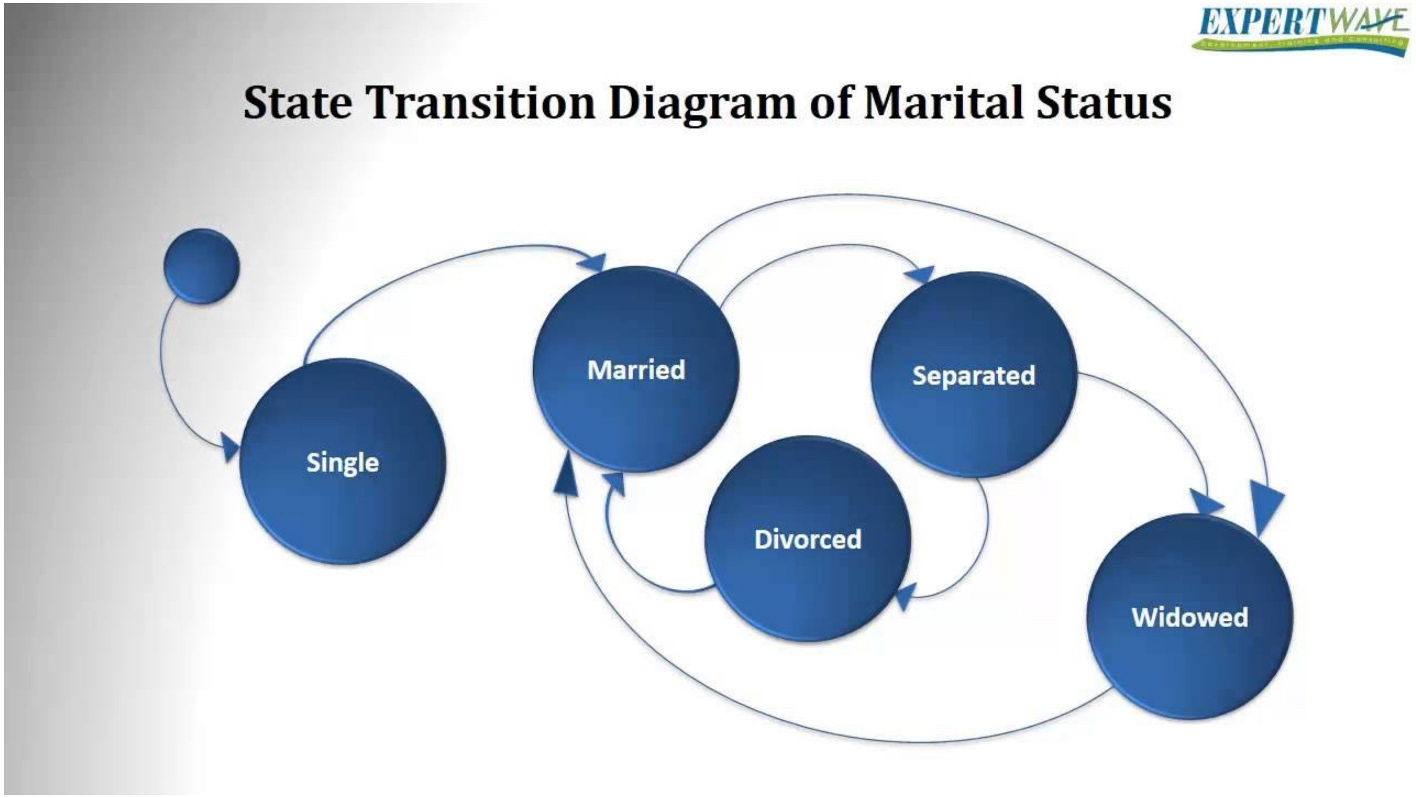
Exercise:

If you are a new customer opening a credit card account, you will get a 15% discount on all your purchases today. If you are an existing customer and you hold a loyalty card, you get a 10% discount. If you have a coupon, you can get 20% off (but it can't be used with the 'new customer' discount). Discount amounts are added, if applicable.

	R1	R2	R3	R4	R5	R6	R7	R8
Conditions								
New Customer	T	T	T	T	F	F	F	F
Loyalty Card	T	T	F	F	T	T	F	F
Coupon	T	F	T	F	T	F	T	F
Actions								
Discount	--	--	20%	15%	30%	10%	20%	0%

© http://infosharehouse.blogspot.com/2015/11/in-previous-articles-we-learned-about.html

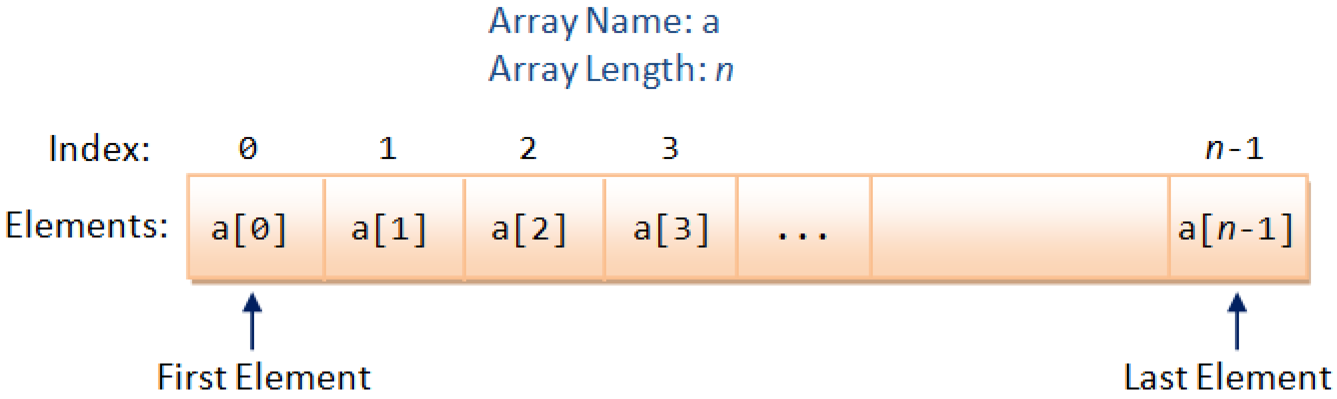
ISTQB State Transition Diagrams:



© https://www.youtube.com/watch?v=SQGIr3AFLzE

Arrays

arrayName[firstElement, second, third]



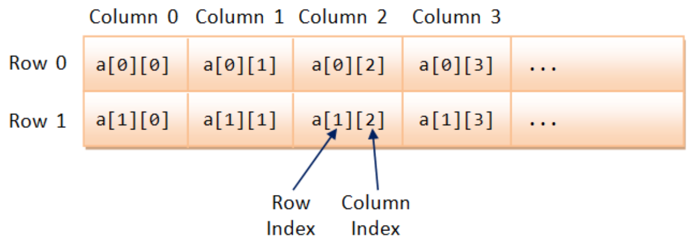
Array Name: a
Array Length: *n*

Index:	0	1	2	3			*n*-1
Elements:	a[0]	a[1]	a[2]	a[3]	...		a[*n*-1]

First Element

Last Element

Also multidimensional arrays:

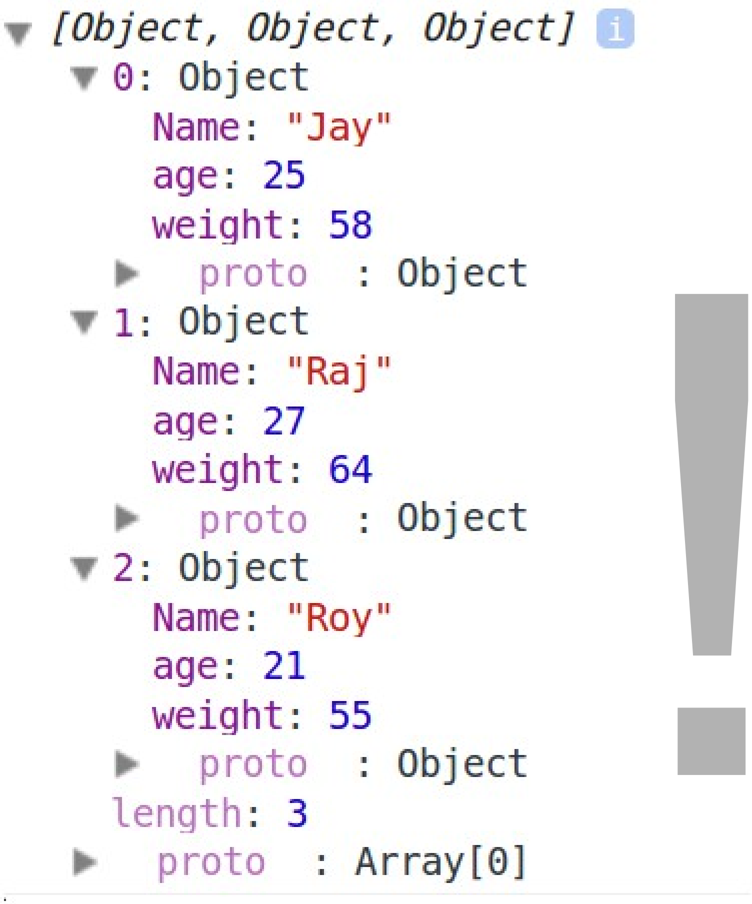


	Column 0	Column 1	Column 2	Column 3	
Row 0	a[0][0]	a[0][1]	a[0][2]	a[0][3]	...
Row 1	a[1][0]	a[1][1]	a[1][2]	a[1][3]	...

© http://searchmicroservices.techtarget.com/definition/data-type

Objects

objectName{
param1:”value”;
param2:”value”
}



```
▼ [Object, Object, Object] i
  ▼ 0: Object
      Name: "Jay"
      age: 25
      weight: 58
    ▶   proto   : Object
  ▼ 1: Object
      Name: "Raj"
      age: 27
      weight: 64
    ▶   proto   : Object
  ▼ 2: Object
      Name: "Roy"
      age: 21
      weight: 55
    ▶   proto   : Object
    length: 3
  ▶   proto   : Array[0]
```

© https://codehandbook.org/javascript-json-array-revisited/

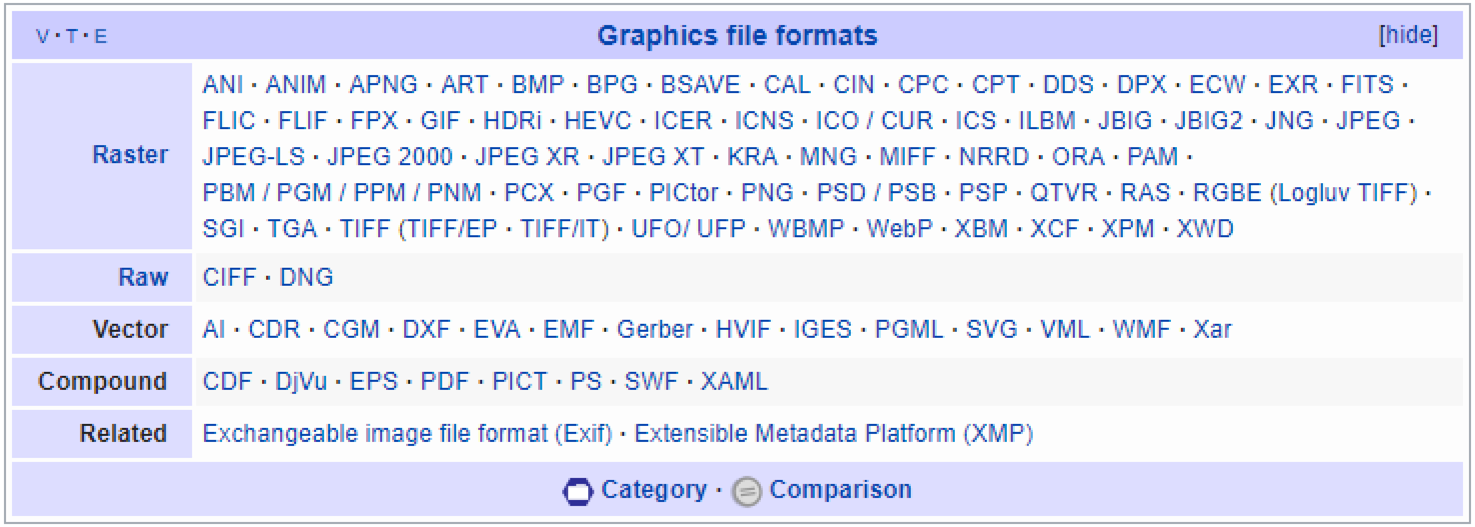
File format

A file format is a standard way that **information is encoded for storage in a computer file**. **It specifies how bits are used to encode information in a digital storage medium.** File formats may be either proprietary or free and may be either unpublished or open.

Some file formats are designed for very particular types of data: **PNG files**, for example, store bitmapped images using lossless data compression. Other file formats, however, are designed for storage of several different types of data: the Ogg format can act as a container for different types of multimedia including any combination of audio and video, with or without text (such as subtitles), and metadata. A **text file** can contain any stream of characters, including possible control characters, and is encoded in one of various character encoding schemes. Some file formats, such as **HTML**, **scalable vector graphics**, and the source code of computer software are text files with defined syntaxes that allow them to be used for specific purposes.

© https://en.wikipedia.org/wiki/File_format

Images files format



V · T · E **Graphics file formats** [hide]

Raster	ANI · ANIM · APNG · ART · BMP · BPG · BSAVE · CAL · CIN · CPC · CPT · DDS · DPX · ECW · EXR · FITS · FLIC · FLIF · FPX · GIF · HDRi · HEVC · ICER · ICNS · ICO / CUR · ICS · ILBM · JBIG · JBIG2 · JNG · JPEG · JPEG-LS · JPEG 2000 · JPEG XR · JPEG XT · KRA · MNG · MIFF · NRRD · ORA · PAM · PBM / PGM / PPM / PNM · PCX · PGF · PICtor · PNG · PSD / PSB · PSP · QTVR · RAS · RGBE (Logluv TIFF) · SGI · TGA · TIFF (TIFF/EP · TIFF/IT) · UFO/ UFP · WBMP · WebP · XBM · XCF · XPM · XWD
Raw	CIFF · DNG
Vector	AI · CDR · CGM · DXF · EVA · EMF · Gerber · HVIF · IGES · PGML · SVG · VML · WMF · Xar
Compound	CDF · DjVu · EPS · PDF · PICT · PS · SWF · XAML
Related	Exchangeable image file format (Exif) · Extensible Metadata Platform (XMP)

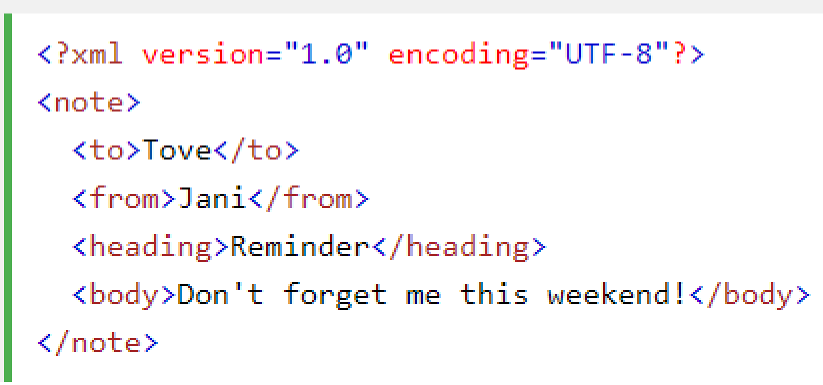
Category · Comparison

© https://vector-conversions.com/vectorizing/raster_vs_vector.html & https://en.wikipedia.org/wiki/Vector_graphics

Data files format

XML

- stands for eXtensible Markup Language, was designed to store and transport data. Designed to be both human- and machine-readable.



```
<?xml version="1.0" encoding="UTF-8"?>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

//[TODO]

try it with editor and browsers to build your own data set:

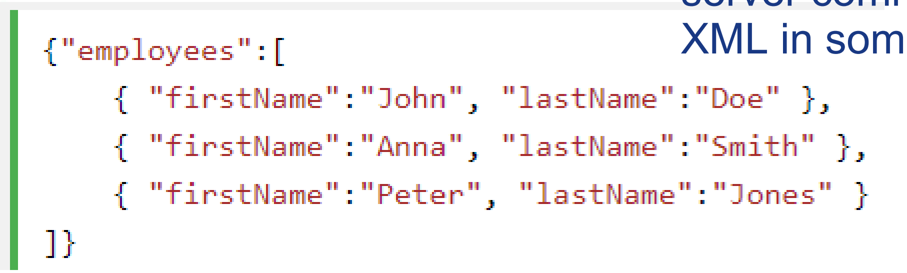
```
<?xml version="1.0" encoding="UTF-8"?>
<note>
  <to>Group</to>
  <from>W2BUSINESS course</from>
  <heading>Reminder</heading>
  <body>Don't forget to do home task!</body>
</note>
```

© https://en.wikipedia.org/wiki/XML & https://www.w3schools.com/xml/default.asp

Data files format

JSON

JavaScript Object Notation is an open-standard file format that uses human-readable text to transmit data objects consisting of attribute–value pairs and array data types (or any other serializable value). It is a very common data format used for asynchronous browser–server communication, including as a replacement for XML in some AJAX-style systems.



```
{"employees":[
    { "firstName":"John", "lastName":"Doe" },
    { "firstName":"Anna", "lastName":"Smith" },
    { "firstName":"Peter", "lastName":"Jones" }
]}
```

//[TODO]

try it with editor and browsers to build your own data set:

```
{"employees":[
    { "firstName":"John", "lastName":"Doe" },
    { "firstName":"Anna", "lastName":"Smith" },
    { "firstName":"Peter", "lastName":"Jones" }
]}
```

© https://www.w3schools.com/js/js_json_intro.asp

JSON vs XML

JSON

```
{"employees":[
    { "firstName":"John",
"lastName":"Doe" },
    { "firstName":"Anna",
"lastName":"Smith" },
    { "firstName":"Peter",
"lastName":"Jones" }
]}
```

XML

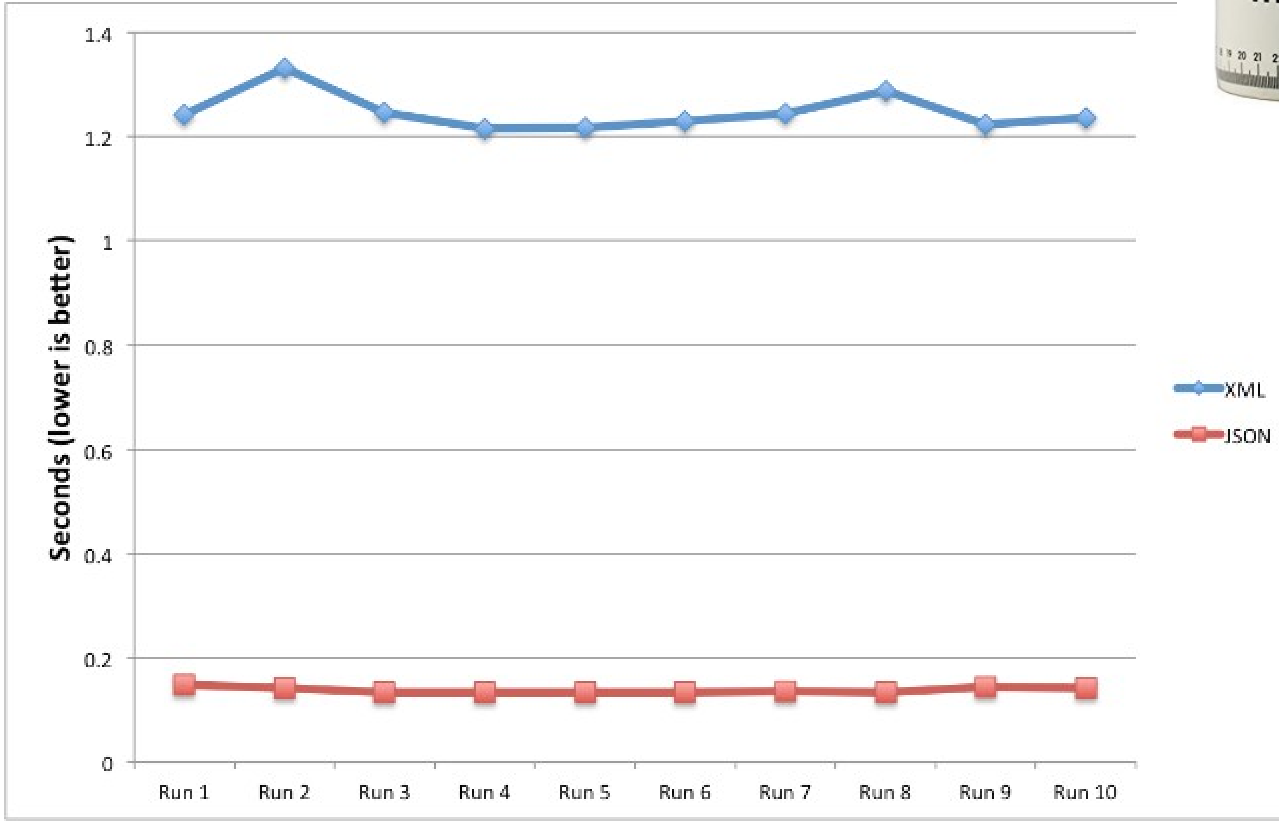
```
<employees>
    <employee>
        <firstName>John</firstName>
<lastName>Doe</lastName>
    </employee>
    <employee>
        <firstName>Anna</firstName>
<lastName>Smith</lastName>
    </employee>
    <employee>
        <firstName>Peter</firstName>
<lastName>Jones</lastName>
    </employee>
</employees>
```

!

© https://www.w3schools.com/js/js_json_xml.asp

JSON vs XML

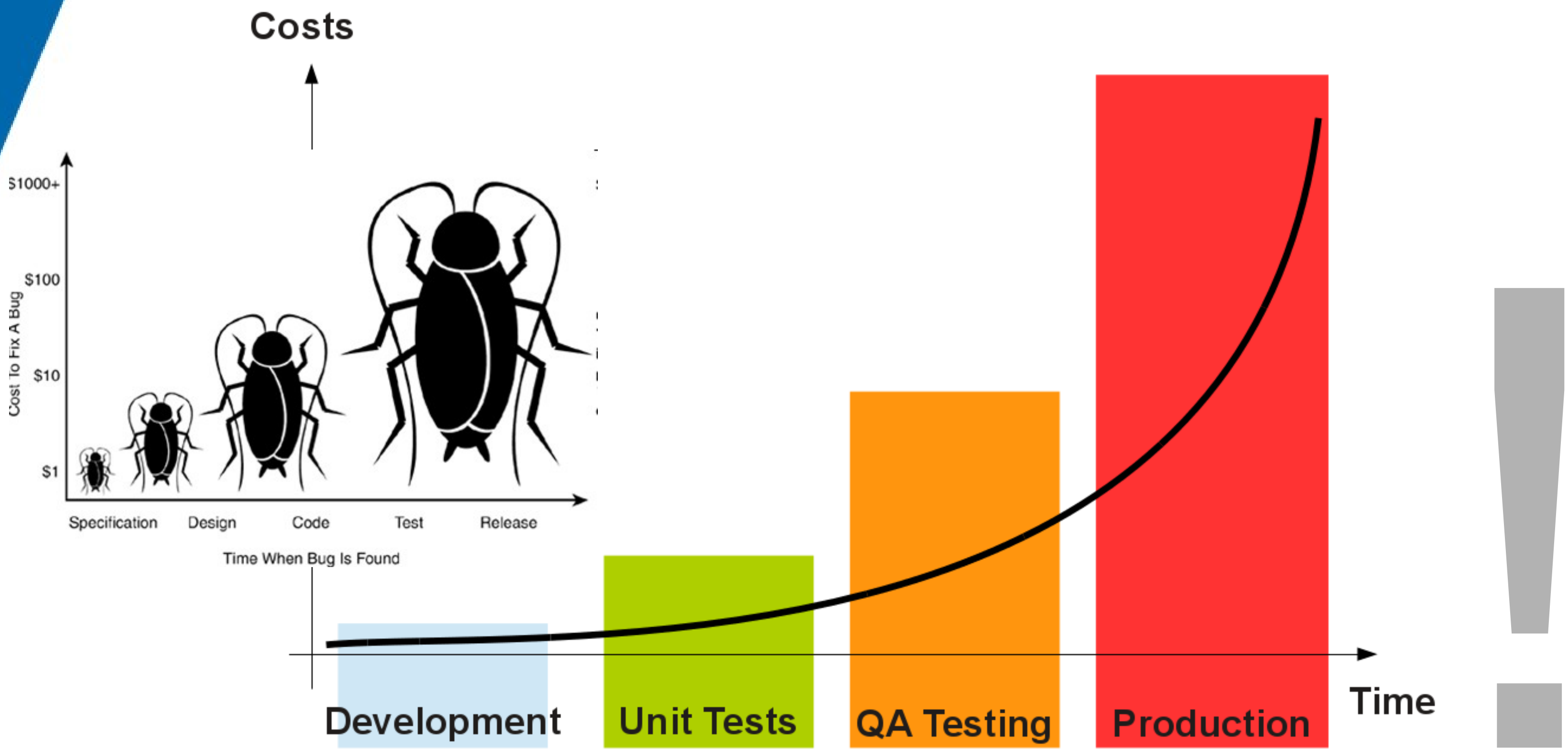
Size matters



!

© https://www.infragistics.com/community/blogs/b/torrey-betts/posts/mobile-performance-testing-json-vs-xml & https://www.amazon.co.uk/Size-Matters-Giant-Mug-Novelty/dp/B007N1JT10

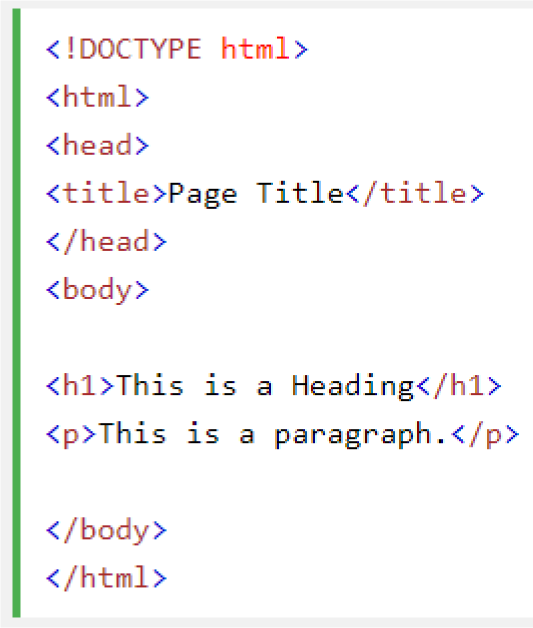
Software Development Costs: Bug fixing



© https://blog.pdark.de/2012/07/21/software-development-costs-bugfixing/ & http://systemsemantics.com/2014/08/the-cost-of-bugs/

HTML - Hypertext Markup Language

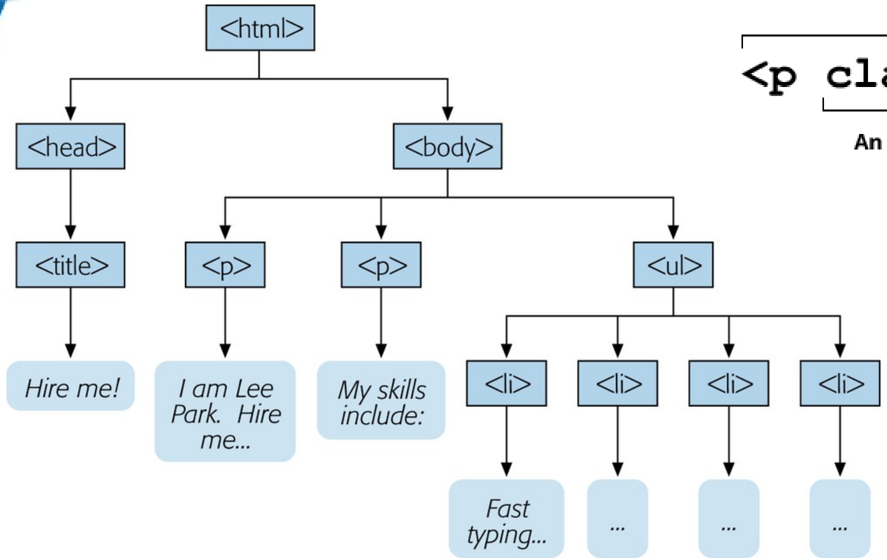
is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receive HTML documents from a web server or from local storage and render them into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.



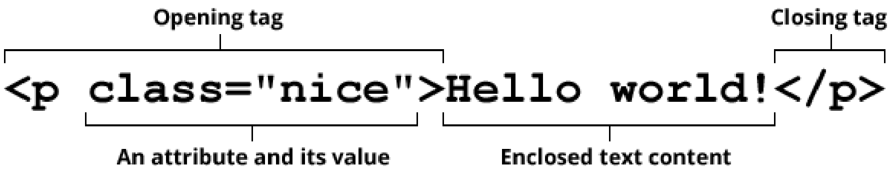
```html
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

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

</body>
</html>
```

© https://www.w3schools.com/html/

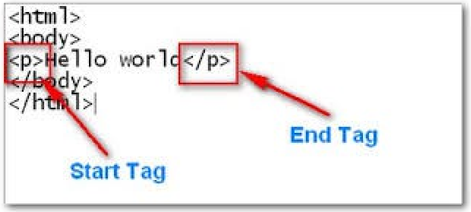
HTML – elements



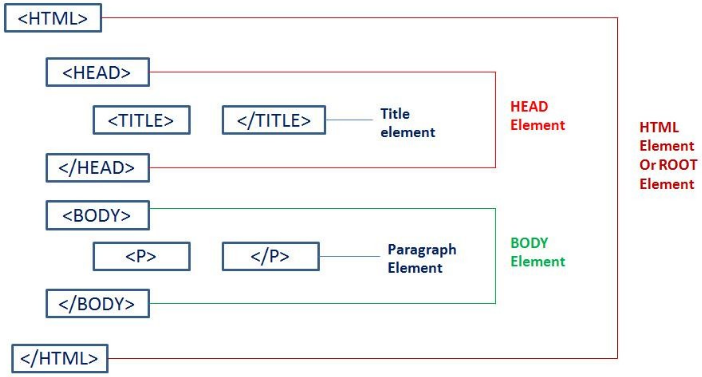
Anatomy of an HTML element



```
<p class="nice">Hello world!</p>
```



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



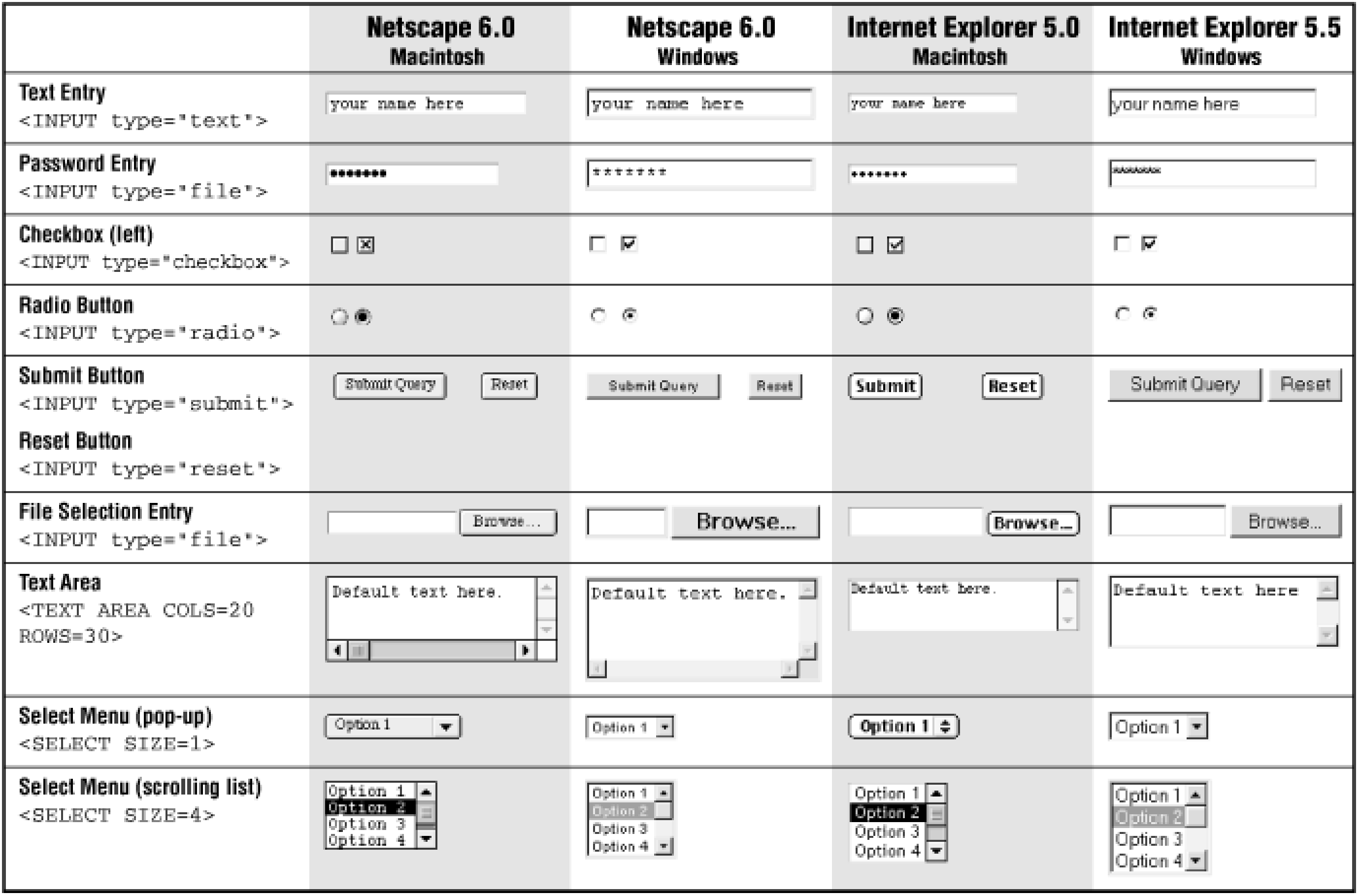
//[TODO]

try it with editor and browsers to build your own web page

!

© https://developer.mozilla.org/en-US/docs/Glossary/Element & https://www.safaribooksonline.com/library/view/creating-a-website/9781449374525/ch01.html

HTML – web form elements



	Netscape 6.0 Macintosh	Netscape 6.0 Windows	Internet Explorer 5.0 Macintosh	Internet Explorer 5.5 Windows
Text Entry `<INPUT type="text">`	your name here	your name here	your name here	your name here
Password Entry `<INPUT type="file">`	•••••••	*******	•••••••	*******
Checkbox (left) `<INPUT type="checkbox">`	☐ ☒	☐ ☑	☐ ☑	☐ ☑
Radio Button `<INPUT type="radio">`	○ ●	○ ●	○ ●	○ ●

Submit Button
`<INPUT type="submit">`
Reset Button
`<INPUT type="reset">`
	Submit Query Reset	Submit Query Reset	Submit Reset	Submit Query Reset

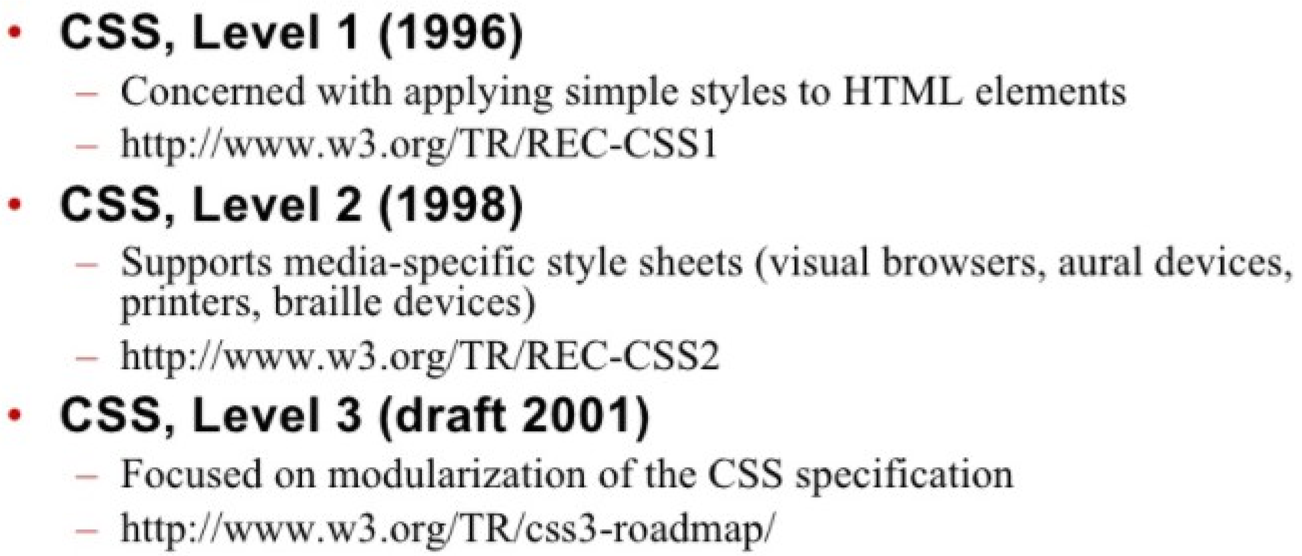
File Selection Entry `<INPUT type="file">`	Browse...	Browse...	Browse...	Browse...
Text Area `<TEXT AREA COLS=20 ROWS=30>`	Default text here.	Default text here.	Default text here.	Default text here
Select Menu (pop-up) `<SELECT SIZE=1>`	Option 1	Option 1	Option 1	Option 1
Select Menu (scrolling list) `<SELECT SIZE=4>`	Option 1 Option 2 Option 3 Option 4	Option 1 Option 2 Option 3 Option 4	Option 1 Option 2 Option 3 Option 4	Option 1 Option 2 Option 3 Option 4

//[TODO]

try it with editor and browsers to build your own web page

© http://www.e-reading.life/htmbook.php/orelly/web2/wdesign/ch15_06.htm

CSS - Cascading Style Sheets

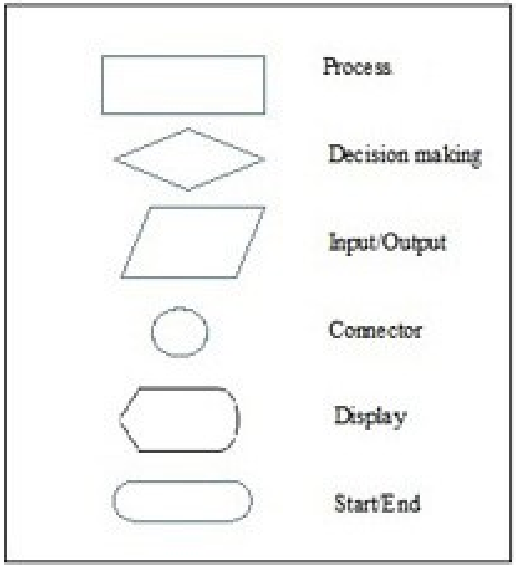


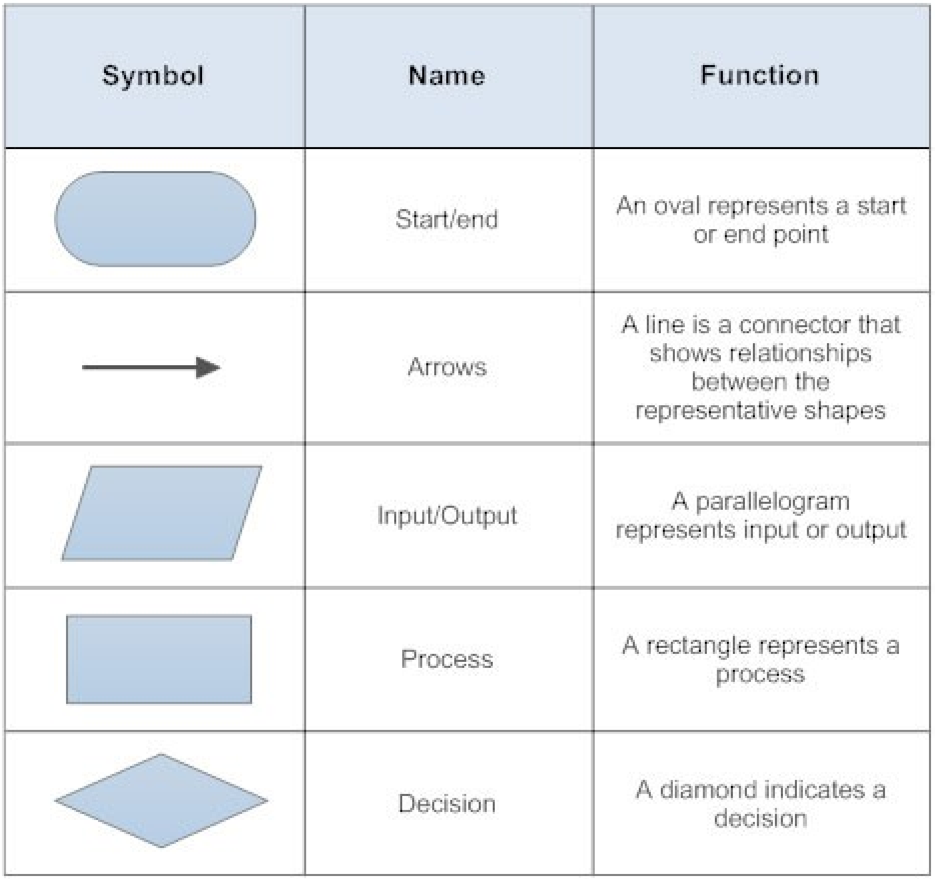
- **CSS, Level 1 (1996)**
  - Concerned with applying simple styles to HTML elements
  - http://www.w3.org/TR/REC-CSS1
- **CSS, Level 2 (1998)**
  - Supports media-specific style sheets (visual browsers, aural devices, printers, braille devices)
  - http://www.w3.org/TR/REC-CSS2
- **CSS, Level 3 (draft 2001)**
  - Focused on modularization of the CSS specification
  - http://www.w3.org/TR/css3-roadmap/

CSS - Cascading Style Sheets

```
<!DOCTYPE html>
<html>
<head>
        <style>
                body {background-color: lightblue;}
                h1 {color: white; text-align: center;}
                p {font-family: verdana; font-size: 20px;}
        </style>
</head>
<body>
<h1>My First CSS Example</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

Program algorithm elements



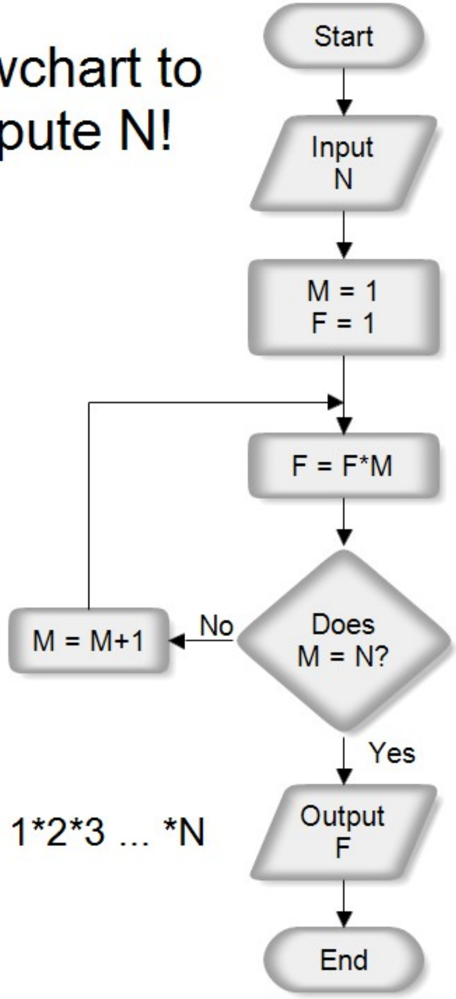


Symbol	Name	Function
⬭	Start/end	An oval represents a start or end point
→	Arrows	A line is a connector that shows relationships between the representative shapes
▱	Input/Output	A parallelogram represents input or output
▭	Process	A rectangle represents a process
◇	Decision	A diamond indicates a decision

© https://study.com/academy/lesson/flowchart-symbols-in-programming-definition-functions-examples.html & https://www.smartdraw.com/flowchart/flowchart-symbols.htm

Factorial flowchart

A Flowchart to Compute N!



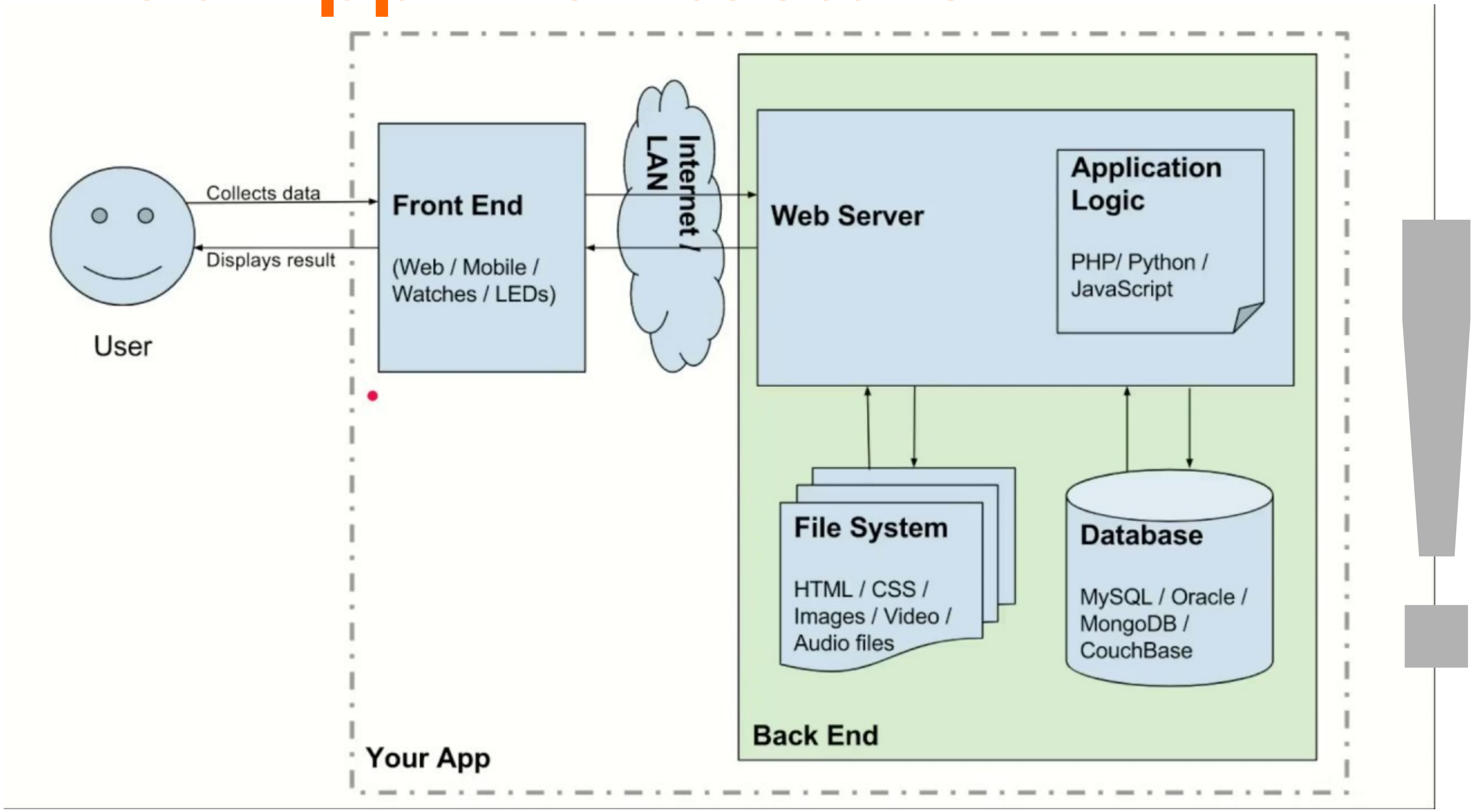
F = 1*2*3 ... *N

!

© https://www.rff.com/n_factorial_flowchart.php

Software Development basics

Do you remember this?
Web App Architecture

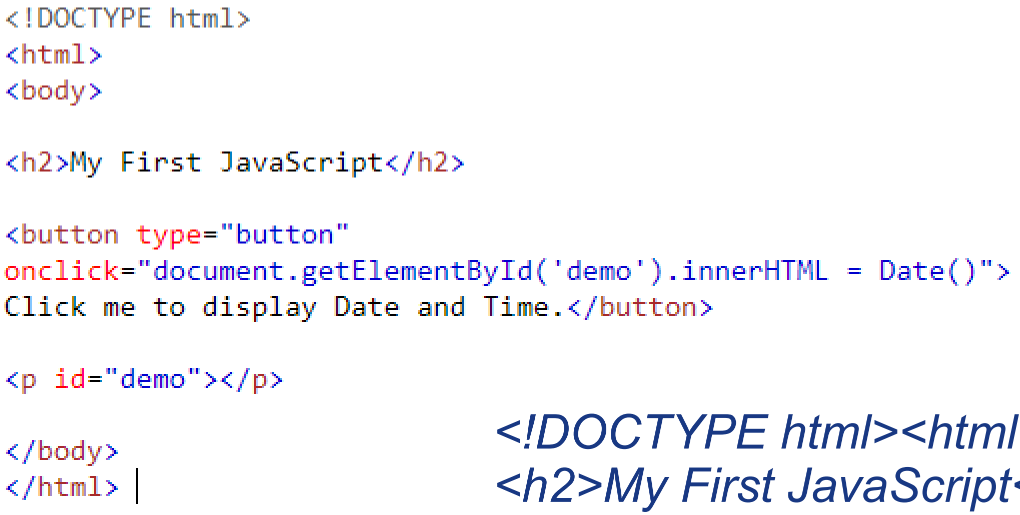


© https://magora-systems.com/mobile-software-development-for-newbies/

JavaScript

JS

- JavaScript, often abbreviated as JS, is a high-level, interpreted programming language. It is a language which is also characterized as dynamic, weakly typed, prototype-based and multi-paradigm.



```
<!DOCTYPE html>
<html>
<body>

<h2>My First JavaScript</h2>

<button type="button"
onclick="document.getElementById('demo').innerHTML = Date()">
Click me to display Date and Time.</button>

<p id="demo"></p>

</body>
</html>
```

//[TODO]

try it with editor and browsers to build your own simple FE code

```
<!DOCTYPE html><html><body>
<h2>My First JavaScript</h2>
<button type="button"
onclick="document.getElementById('demo'
).innerHTML = Date()">
Click me to display Date and Time.</button>
<p id="demo"></p>
</body></html>
```

!

© https://en.wikipedia.org/wiki/XML & https://www.w3schools.com/xml/default.asp

JavaScript

Simple examples

```
var x; // defines the variable x and assigns to it the special value "undefined" (not to be confused with an undefined value)
var y = 2; // defines the variable y and assigns to it the value 2
var z = "Hello, World!"; // defines the variable z and assigns to it a string entitled "Hello, World!"
var cars = ["Saab", "Volvo", "BMW"]; // defines array
```

Tips

```
console.log("Hello World!");
```

A simple recursive function:

```
function factorial(n) {
    if (n === 0 || n === 1) {
        return 1;  // 0! = 1! = 1
    }
    return n * factorial(n - 1);
}

factorial(3); // returns 6
```

!

//[TODO]

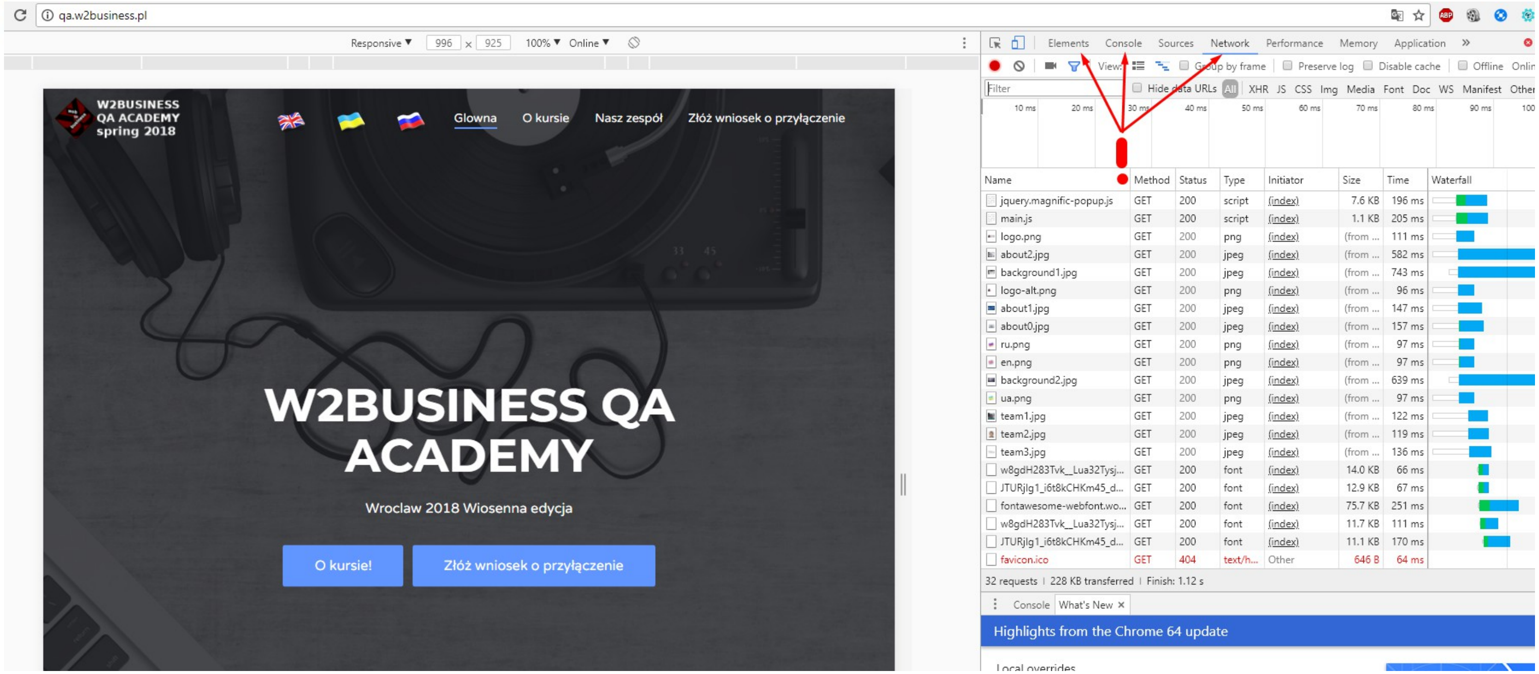
try it with editor and browsers to build your own simple FE code

© https://en.wikipedia.org/wiki/XML & https://www.w3schools.com/xml/default.asp

JavaScript

Browser console

F12 and enjoy it:



Name	Method	Status	Type	Initiator	Size	Time	Waterfall
jquery.magnific-popup.js	GET	200	script	(index)	7.6 KB	196 ms	
main.js	GET	200	script	(index)	1.1 KB	205 ms	
logo.png	GET	200	png	(index)	(from ...	111 ms	
about2.jpg	GET	200	jpeg	(index)	(from ...	582 ms	
background1.jpg	GET	200	jpeg	(index)	(from ...	743 ms	
logo-alt.png	GET	200	png	(index)	(from ...	96 ms	
about1.jpg	GET	200	jpeg	(index)	(from ...	147 ms	
about0.jpg	GET	200	jpeg	(index)	(from ...	157 ms	
ru.png	GET	200	png	(index)	(from ...	97 ms	
en.png	GET	200	png	(index)	(from ...	97 ms	
background2.jpg	GET	200	jpeg	(index)	(from ...	639 ms	
ua.png	GET	200	png	(index)	(from ...	97 ms	
team1.jpg	GET	200	jpeg	(index)	(from ...	122 ms	
team2.jpg	GET	200	jpeg	(index)	(from ...	119 ms	
team3.jpg	GET	200	jpeg	(index)	(from ...	136 ms	
w8gdH283Tvk__Lua32Tysj...	GET	200	font	(index)	14.0 KB	66 ms	
JTURjIg1_i6t8kCHKm45_d...	GET	200	font	(index)	12.9 KB	67 ms	
fontawesome-webfont.wo...	GET	200	font	(index)	75.7 KB	251 ms	
w8gdH283Tvk__Lua32Tysj...	GET	200	font	(index)	11.7 KB	111 ms	
JTURjIg1_i6t8kCHKm45_d...	GET	200	font	(index)	11.1 KB	170 ms	
favicon.ico	GET	404	text/h...	Other	646 B	64 ms	

32 requests | 228 KB transferred | Finish: 1.12 s

PHP

- is a server-side scripting language designed for web development but also used as a general-purpose programming language. Originally created by Rasmus Lerdorf in 1994, the PHP reference implementation is now produced by The PHP Group. PHP originally stood for Personal Home Page, but it now stands for the recursive acronym PHP: Hypertext Preprocessor

```
<!DOCTYPE html>
<html>
<body>

<?php
echo "My first PHP script!";
?>

</body>
</html>
```

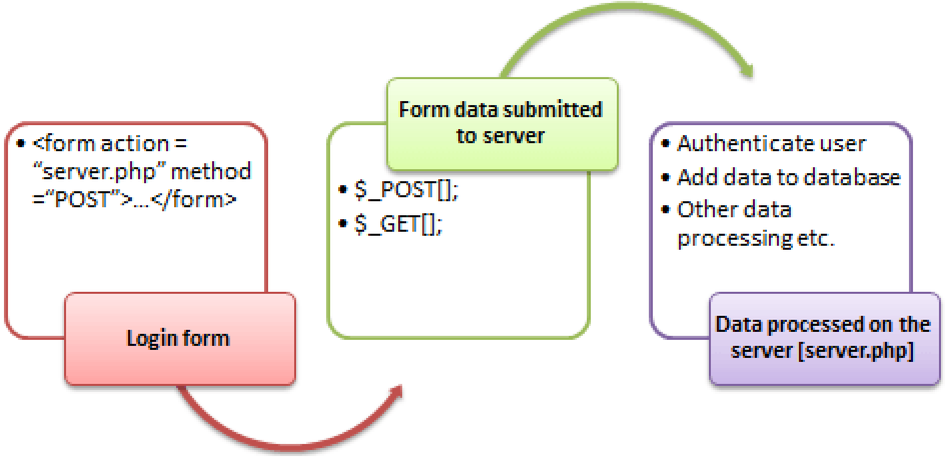
//[TODO]

try it with editor, browsers, XAMPP to build your own simple BE code script

© https://en.wikipedia.org/wiki/XML & https://www.w3schools.com/xml/default.asp

HTML => PHP

Information sent from a form with the POST method is invisible to others (all names/values are embedded within the body of the HTTP request) and has no limits on the amount of information to send.



- <form action = "server.php" method ="POST">...</form>
Login form
Form data submitted to server
- $_POST[];
- $_GET[];
- Authenticate user
- Add data to database
- Other data processing etc.
Data processed on the server [server.php]

//[TODO]

try it with editor and browsers and serwer with PHP (as ex. XAMPP) to build your own simple FE-BE APP

!

```
<html>
<body>

<form action="action.php" method="post">
 <p>Your name: <input type="text" name="name" /></p>
 <p>Your age: <input type="text" name="age" /></p>
 <p><input type="submit" /></p>
</form>

</body>
</html>
```

```
Hi <?php echo htmlspecialchars($_POST['name']); ?>.
You are <?php echo (int)$_POST['age']; ?> years old.
```

© https://www.w3schools.com/php/php_forms.asp & http://php.net/manual/en/tutorial.forms.php

Self-training home tasks:

//[TODO]

0. Use XAMPP to up local host server
1. Get updates from repo https://github.com/AndriiStepura/W2BUSINESS_QA_Academy with command: git bash pull
2. In Asana project assign any tasks from "To Do" with title "Lecture #2 - Homework task theory #WQAA-..." to yourself and set as "In Progress".
   - 2.1. Read ISTQB syllabus 3 and 4 chapters (30-44 pages)
   - 2.2. Fill the answers in the file "2.Software_Development_basics.xls" and attach it to task and set as "Ready To Review"

3. Assign any another team member's task from "Ready to Review" to yourself and change status to "In Review".
   - 3.1. Create HTML file with form based on template "2.html_form_template.html" for 10 questions.
   - 3.2. Fill form according to input from xls file from ticket answers, then click "submit and check answers set".
   - 3.3. According to received result:
     - If 1 incorrect answer – assign this task to previous persons and set the task as "Ready for Tests"
     - If more than one incorrect answer check your form outputs and server side inputs OR assign this task to previous person, add as comments the result of checking and set the task as "To Do".

4. Assign any another team member's task from "Ready for Tests" to yourself and set to "In Test" AND assign task from "To Do" with title "Lecture #2 - Homework task PRACTICE #WQAA-..." to yourself and set as "In Progress".
   - 4.1. Check theory answers and created AJAX file with answers according to "2.data_template.json" and fill the file according to answers in xls file from ticket.
   - 4.2. Check all your answers in json by file "2.json_checker.html" (both files should be in the same folder).

6. Check that you are not assigned to any task with status "To Do".

=== Additional tasks for advanced:

7. Use HTML5 parameters to validate web form inputs
8. Use JavaScript to validate form from FE side
9. Use PHP to validate form from BE side
10. Create new task "2.5.Software_Development_basics_ADD#WQAA-{integer}" and set it to review.
11. Get any task from ready to review AND test with title "2.5.Software_Development_basics_ADD#WQAA-{integer}" and check this code.

Gratitude:

Thanks for review:

http://w2business.pl/
W2BUSINESS
QA academy

Thanks for tech background:

FUNDACJA
UKRAINA

https://www.linkedin.com/company/10948809/

https://www.facebook.com/fundacja.ukraina/