college CD

Scripting languages II (LS2)

Class 03

Top programming languages & frameworks (2019)

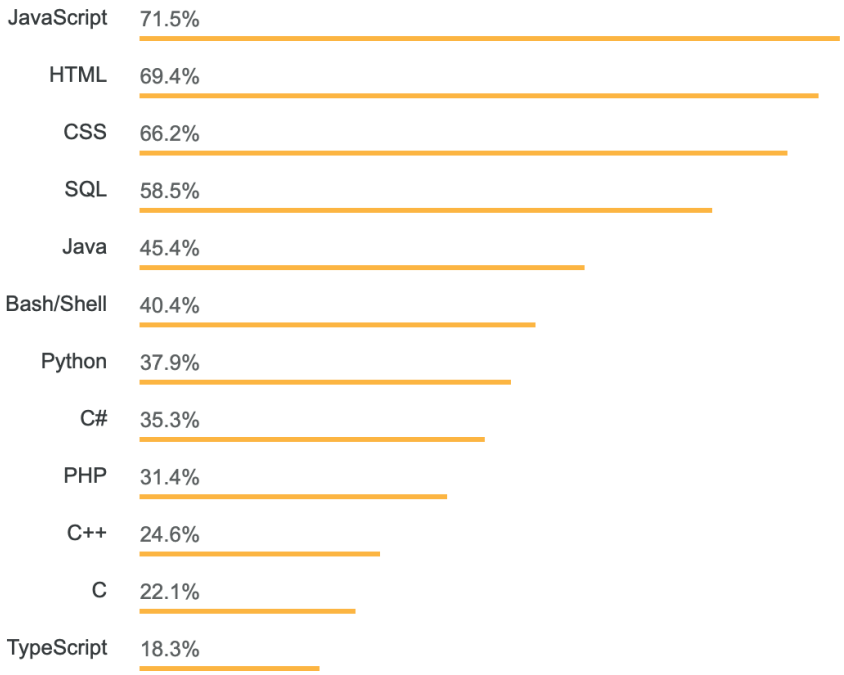
Every language has its own pros and cons. Therefore, the choice of a language should focus on your specific needs. Over popularity, there are many factors to consider when comes the time to choose a scripting language.

JavaScript

Most probably due to its compatibility with all browsers, JavaScript is the most popular language and it is preferred by most full-stack software developers nowadays. It is the leader when it comes to front-end development and it can be used to very conveniently develop interactive web pages.

JavaScript may also now be used in server-side through node.js replacing languages such as PHP.

In 2018, *Stack Overflow* surveyed 100,000 developers to find the most preferred language: Javascript was ranked number one, followed by HTML, CSS, SQL, and others.



Python

Python has gain more in popularity than any other language in the last few years mostly because of the rise of Artificial intelligence (it is the preferred language for creating AI or machine learning based web & mobile apps), and, also, because of the elimination of semicolon at the end of the statement. In many ways, it is similar and different from other programming languages.

The fastest growing python package is Pandas (introduced in 2011), a multi-purpose language which can be used for data science as well as web designing and it is gradually replacing Javascript as a teaching language in a growing number of institutes.

Java

Considered as one of the most stable languages, Java, a cross-platform programming language, has been at the peak of the programming industry for the past 20 years and is widely used for building enterprise scale web applications (including Android mobile apps).

PHP

Standing for *Hypertext Preprocessor*, PHP is a popular scripting language introduced 1995 and that kept on evolving throughout these years, leaving no place to any new languages. Most of the available frameworks are free and provide strong security features. It offers many advantages such as libraries and modules which assure dynamic software development. According to a survey by w3techs.com, PHP is considered as the best server-side programming language.

SQL

Standing for *Structured Query Language*, and used for securely managing data structures and information stored in databases, SQL is incredibly popular mainly because its great features and that it allows faster retrieving of a large number of database records. According to the Evans Data Corporation, out of the 19 million developers in the world, 7 million are using SQL.

Ruby

Developed in Japan (mid-1990s), Ruby is a dynamic language constructed in order to simplify the programming task (making it more fun). The main reason of its popularity is the Ruby of rails framework which is a full-stack web development framework.

C/C++

Parent of many modern languages (including Java), and introduced in the 1970s, C still manages to be in the top software writing languages. More dynamic language than C, C++ code is type-checked before it gets actually executed. A major difference between these languages is that C does not supports encapsulation, polymorphism, and inheritance while C++ does.

Swift

Considered a beginners' friendly language developed by Apple Inc (introduced in 2014), Swift is the best choice for the development of native iOS or MAC OS app. Influenced by Python and ruby, it is faster and more secure than both of them.

Javascript Libraries

There are a billion of web pages on the Internet, and many of them are interactive many ways. Think of how many websites have you seen that use a slideshow.

Now, it's possible that everyone of those web developers wrote their own JavaScript code to build a slideshow, but a lot of them probably re-used the same code. A programmer should re-use existing code whenever it's possible, so no time is wasted in rewriting codes another programmer has already written. Doing so, in JavaScript, is done by using a library, which is a JavaScript file that contains a bunch of functions accomplishing various useful tasks.

The best way to know what functions are available consists into reading the framework's documentation. Most libraries have documentation including a list of available functions or a real-world example.

jQuery

jQuery is a lightweight *write less do more* JavaScript library which purpose is to make JavaScript easier to use on websites. This framework takes a lot of common tasks requiring many lines of JavaScript code, and wraps them into methods that can be called using a single line of code. It also simplifies a lot of the complicated tasks from JavaScript, like AJAX calls and DOM manipulation.

jQuery library contains the following features:

- HTML/DOM manipulation
- CSS manipulation
- HTML event methods
- Effects and animations
- AJAX
- etc.

Why use jQuery?

There are many other more recent and modern JavaScript expendable frameworks available (such as Angular, React or Vue), but many of the most important websites on the web use jQuery (Google, Microsoft, IBM, Netflix...). According to a survey by Stack Overflow (2018) about 48% of developers still use jQuery and that may have to do with the fact the very popular Bootstrap framework depends on it.

jQuery : the basics

jQuery consist in another faster and more efficient way to code JavaScript. In order to use this framework, it first needs to be accessible in the HTML document.

Adding jQuery to Your Web Pages

There are two ways to start using jQuery in a web document :

Download the jQuery library

The library may be downloaded from jQuery.com and the folders and files needs to be placed in the web site's root. The jQuery library consists in a single JavaScript file used within the HTML <script> tag which needs to be placed inside the <head> section.

```
<head>
        <script src="/jquery/jquery-3.4.1.min.js"></script>
</head>
```

Two versions of jQuery are available which can be downloaded from jQuery.com :

- **Production version :**
  This is for your live website because it has been minified and compressed.

- **Development version :**
  This is for testing and development (uncompressed and readable code).

Include jQuery from a CDN

jQuery CDN is hosted by both Google and Microsoft).

```
<head>
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/
        jquery.min.js"></script>
</head>
```

Coding in jQuery

Using jQuery

After having linked the library to the HTML document, it is time to start coding scripts. Just like for JavaScript, the codes must be placed within ***<script></script>*** tags, usually after the contents or in an external ***.js*** file that would be linked to the document.

You can see jQuery as a JavaScript function which can be coded in two different ways : using the word jquery or using a dollar sign. Bot ways indicate jQuery is being used.

```
<script>
        jquery();
        $();
</script>
```

Targeting an element

In order to target a specific element, the element which is targeted will be indicated within the function's parenthesis. different methods may then be used, separated by periods.

```
<script>
        $(document).method();
</script>
```

document ready handler

The first thing that needs to be done is to wrap all the jQuery coding within the ***document ready*** handler. Almost everything coded in jQuery needs to be contained in this handler which does two things:

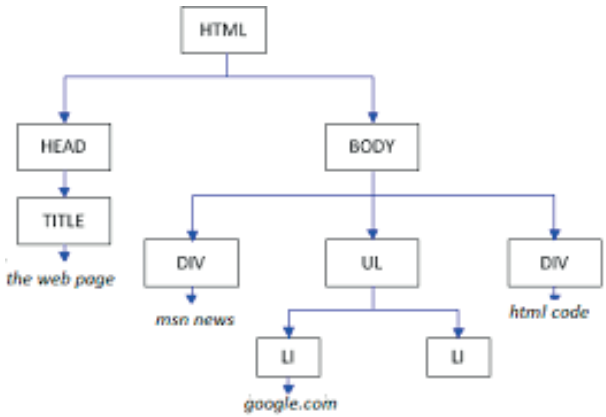
- It ensures that the web document has been fully downloaded before running the jQuery codes. If any elements being accessed would actually be inexistent when running the codes, the script would return an error.
- It ensures that the code is unobtrusive, that is, it's separated from content (XHTML) and presentation (CSS).

```
<script>
        $(document).ready(function() {
         // all jQuery code goes here
});
</script>
```

DOM Manipulation

The Document Object Model (DOM) is a programming interface for HTML and XML documents. It represents the document as ***nodes*** and ***objects*** so that programs can target and change the structure, the style, and specific content. The DOM represents.

A Web page is a document which can either be displayed in the browser window or as the HTML source. But it is the same document in both cases. The Document Object Model (DOM) represents that same document so it can be manipulated. The DOM is an object-oriented representation of the web page, which can be modified with a scripting language such as JavaScript.



Targeting and manipulating elements

One of the most important aspects of JavaScript (so of jQuery), is manipulation of the DOM which allows the developer to target and manipulate specific elements of a document. DOM navigation and manipulation using standard JavaScript can be quite difficult, but jQuery helps greatly with a collection of DOM related methods, making it much more easier.

jQuery selectors

Once jQuery has been included in a document, the first thing to know is how to select HTML elements. jQuery consists in a function that can be passed a series of arguments, the most common being a string containing a CSS selector. These selectors are used in order to target HTML elements to apply certain behaviours.

```
$("div");          // selects all <div> tags
$("#myElement");   // selects tags using ID "myElement"
$(".myClass");     // selects tags using class "myClass"
$("p#myElement");  // selects <p> tags using ID "myElement"
$("p a.navigation"); // selects <a class="navigation"> nested in <p>
```

jQuery supports the use of all CSS selectors :

```
$("p > a");              // <a> that are direct children of paragraphs
$("input[type=text]");   //  inputs of a specified type
$("a:first");            // first <a> on the page
$("p:odd");              // all odd numbered paragraphs
$("li:first-child");     // each list item that's the first child in its list
```

jQuery also allows the use of its own custom selectors :

```
$(":animated");          // elements currently being animated
$(":button");            // any button elements
$(":radio");             // radio buttons
$(":checkbox");          // checkboxes
$(":checked");           // checkboxes or radio buttons that are selected
$(":header");            // header elements (h1, h2, h3, etc.)
```

To check the entire list of jquery selectors, visit :
https://api.jquery.com/category/selectors/

Manipulating an element

After targeting an element, it is possible to manipulate it. To write « Hello world! » within a container using the ID « test1 ». We would first need to target the element, and then use a method to manipulate it.

```
<div id="test1"> </div>
<script type="text/javascript">
        $("#test1").text("Hello world!");
</script>
```

Method chaining

One advantage of jQuery is the fact that most of the methods returns a jQuery object which can then be used to call another method. This allows to do command chaining, where multiple methods can be performed on the same set of elements, which keeps from having to find the same elements more than once.

```
<div id="test2"></div>
<script type="text/javascript">
        $("#test1").text("Hello, world!").css("color", "blue");
</script2
```

Manipulation using events

In order to manipulate an element on an event, it is first needed to target the element and then use a specific event handler which will contain the desired manipulations wrapped in a function.

```
<h1>Test 3</h1>
<button id="test3">Click here</button>
<script type="text/javascript">
$("#test3").click(function() {
        $("h1").css("color", "red");
});
</script>
```

Explanation:
In the example above, the title using <h1> changes to red when the button using ID test3 is clicked.

jQuery includes different event handlers:

blur, focus, hover, keydown, load, mousemove, resize, scroll, submit, select.

Hiding and showing contents

hide( )

Similar to the CSS property *display:none*, hide() method hides the selected elements.

show( )

Used to show hidden content using *display:none*, for instance.

Toggle( )

Allow to alternatively show and hide content.

Note:

Both hide(), show() and toggle() methods can use parameter such as fast, slow or milli-seconds creating some sort of transition