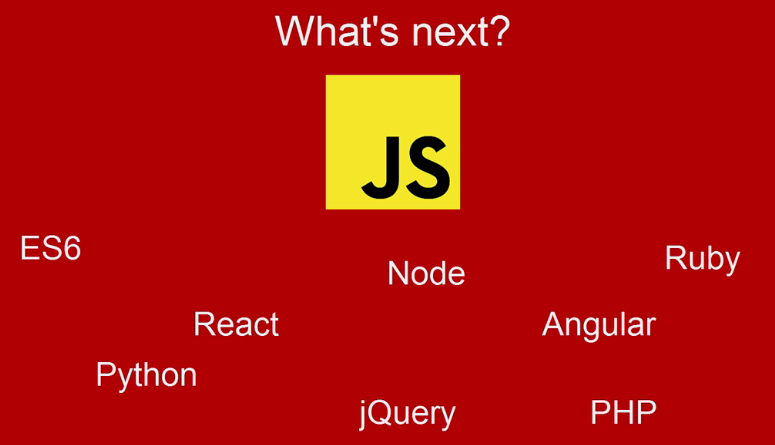
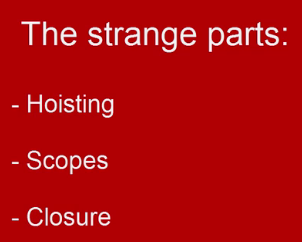
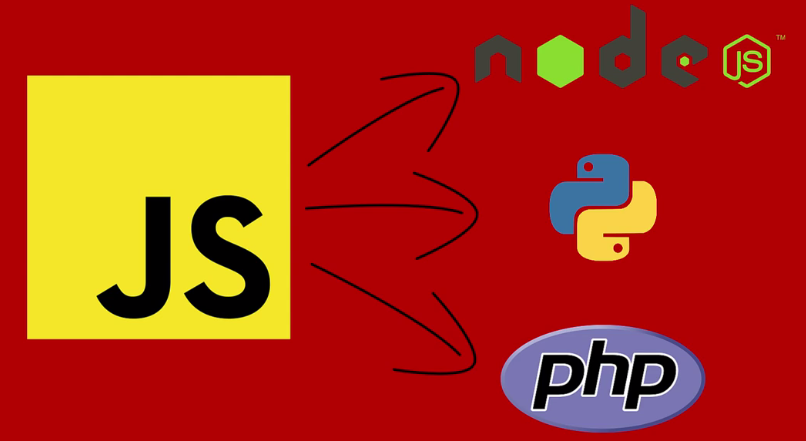
JavaScript 1

**Kalob & Andrei**

**1.1 JavaScript Essentials**

* JavaScript Essentials:



* What is JavaScript: It is scripting language. Similar to C but not same as Java.
* Some of the powers JavaScript has are:
* Changing page styling instantly.
* AJAX'ing other pages.
* Hiding showing elements.
* Taking in user input
* Doing math
* Telling time and dates
* and more...
* JavaScript Internal vs. External
* Internal:
  + In **<head>** inside **<script>**.
  + In **<body>** inside **<script>**, generally we put **<script>** (our JS code) at the end of ***<body>***.
* External: Inside an external JS file. We can link any ***JS*** file using ***<script>*** tag, using "***src***" attribute. We link those JS files inside **<head>** or end of the ***<body>***, these appearance depends on various situation.
  + Appearance of **<script>**, depends on ***HTML*** and ***CSS*** that JS file is used.

|  |  |
| --- | --- |
| Internal | External |
| <!DOCTYPE html>  <html lang="en">      <head>          <meta charset="utf-8">          <title>BoilerPlate</title>          <link>          <script>              // JS code goes here          </script>      </head>        <body>          <p>Hello</p>          <div>              <!-- Other codes -->          </div>            <script>              // JS code goes here          </script>      </body>  </html> | <!DOCTYPE html>  <html lang="en">      <head>          <meta charset="utf-8">          <title>BoilerPlate</title>          <link>          <script src="JS\_file\_URL\_Link" type="text/javascript"></script>      </head>        <body>          <p>Hello</p>          <div>              <!-- Other codes -->          </div>            <script src="JS\_file\_URL\_Link" type="text/javascript"></script>      </body>  ​  </html> |

* JavaScript comments:

// Single line Comment

/\*Multi line

Comment\*/

* **document.write():** Used to write anything in our HTML page.
* We can print normal text:

document.**write**("My Name is Khan");

* Tags can be used

document.**write**("My Name is <strong>Khan</strong>");

* Display info from the browser ***alert*** & ***confirm***

**alert**("Oy! Where are you goinn?")

**confirm**("do u wanna leave?")

* Using variable, and using condition. We put ***Boolean*** result of ***confirm()*** in variable ***c*** and used it inside condition-if.

var c = **confirm**("Ok or Cancel?");

if(c){

**alert**("You pressed \"ok\" ");

} else{

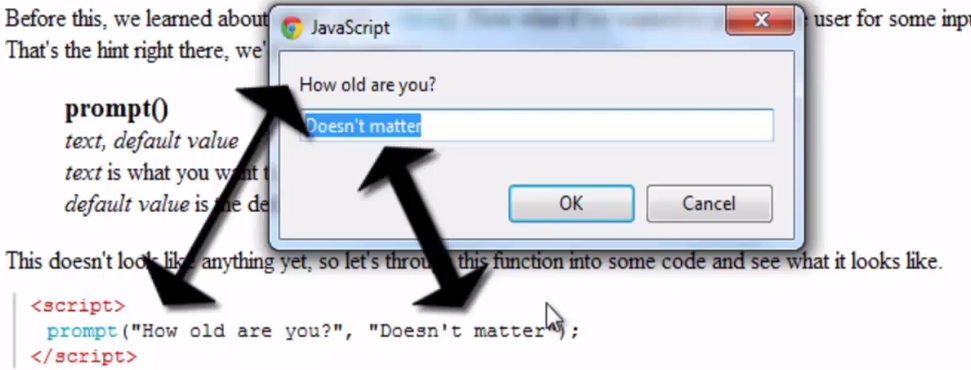
**alert**("You pressed \"Cancel\" ");

}

* ***prompt():*** Prompting the user for information

**prompt(**text, default\_value**)**

* ***text*** is what you want to ask, it’s the question.
* ***default\_value*** is the default value for the form, displayed in the field.



**prompt**("Oy! Where are you goinn?", "City");

* Using given answer from a prompt Storing it to a variable.

/\* Using given answer from a prompt

Storing it to a variable \*/

var pmt = **prompt**("How ld are you?", "");

if(pmt){

**alert**(pmt + " is your age");

} else{

**alert**("No input was typed");

}

**1.2 Programming fundamentals**

***var:*** Declares a variable, optionally initializing it to a value.

***let:*** Declares a block scope local variable, optionally initializing it to a value.

* Before 2015, using the ***var*** keyword was the only way to declare a JavaScript variable.
* The 2015 version of JavaScript (*ES6*) allows the use of the ***const*** keyword to define a *variable that cannot be reassigned*, and the ***let*** keyword to define a variable with restricted scope.

var big = 0;

const grav\_acc = 9.8;

if (grav\_acc > 9.8){

    let to\_space = true;

    return to\_space;

} else{

**alert**("You canot goto space");

}

* Restrictions: Same as other languages.

/\* Declaration rules \*/

var $Name = "My Nair.e"; // this works!

var Name\_2 = "My Nair.e"; // this works!

var \_Name = "My Nair.e"; // this works!

var 2Name = "My Nair.e"; // this does not work!

var #Name = "My Nair.e"; // this does not work!

**1.3 Local & Global Variables**

* Local: Local var are inside a function. Cannot be accessed outside of the function.
* Global: Outside of a function, used by multiple functions.

**1.4 Concatenation**

String + String

String + Number

**1.5 Basic math in JavaScript**

Four basic operations **=, -, /, \***

Modulus **%**

Increment **++**

Decrement **--**

* Short hand assignment:

+=

-=

\*=

etc

// Example

var p = **prompt**("How old are you right now?");

if(p){

    p++;

**alert**("You will be" + p + "next year!");

} else{

**alert**("Sorry, you didn't put a number in");

}

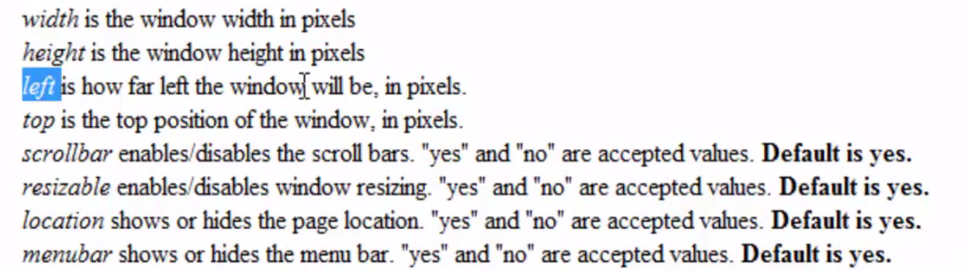
**1.6 Redirecting and opening new windows**

* window.location: With the ***window.location*** variable, we can change the ***location*** of the ***current page***. We can direct or redirect traffic to anywhere we like.
* Be careful not to redirect the page *back to itself* or to a *page* that brings the *user back* to *your page*. This can cause a bad cycle and ultimately force your browser to stop loading the page(s).

// change the location of current page

window.location = "https://google.com";

* window.open: URL, name, options, replace - All param's are1optional
* ***URL*** is what page to open, ie: "http:goo gle.com". If no param is specified aboutblank will open.
* ***name*** is the ***target=""*** attribute. ***\_blank***, ***\_self***, ***\_parent***, ***\_top***, ***\_framename*** are acceptable answers.
* ***options*** are how you want the window to be opened. Eg: "width = 200, height = 400, scrollbar = no"
* Some browsers think they know what's best for you and overwrite the coders demands. All listed together in the same quotations but separated by commas.



* ***replace*** will replace the ***current page*** or create a new page in your history list.
* ***true*** will ***replace*** the current history page
* ***false*** will create a ***new*** history page, ***Default***.

// Openning new window window.opem(p1, p2, p3, p4): URL, name, options, replace

window.**open**(URL = "https://google.com", target = "\_blank", "width = 200, height = 400, scrollbar = no", "true");

* Above is similar to:

<a href="#", target="\_blank"></a>

**1.7 Empty hyperlinks**

* **Using "#":**

<a href="#"></a>

* **Javascript void(0):**

<a href="javacsript: void(0);">Empty Link</a>

            <a href="#", target="\_blank"></a>

            <a href="javacsript: void(0);">Empty Link</a>

* The problem with that, in most browsers, the **#** link will take you to the ***top*** of the ***page***. That's because the ***browser*** is ***looking*** for a ***name*** starting with a ***#***. Remember how we can link to ***#CustomName*** in the same page?
* Why use **void(0)** then?
* To use an event trigger, such as ***oncliclc*** or ***onmousedown*** to activate a ***javascript*** ***function***.

**1.8 String Manipulation**

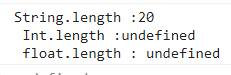
* String: A variable that contains letters.
* Int: As usual, integer numbers.
* Float: Decimal poinr numbers.
* **Var.length:** Find length of a variable, actually we can only find the length of string variables.

var alphabet = "uydgb aiufhpewoifn 0";

var age = 22;

var ExactAge = 22.989080;

console.**log**("String.length :"+alphabet.length+"\n Int.length :"+age.length+"\n float.length : "+ExactAge.length);



* **str.toUpperCase();**

alphabet.**toUpperCase**();

* **str.toLowerCase();**

"Monday Is NOT good".**toLowerCase**();

* Those are good for comparing strings:

/\* Comparing strings \*/

var p = **prompt**("What is the password?", "");

if(p.**toLowerCase**()=="password" && p.length==8) {

**alert**("Access granted!");

    } else {

**alert**(p.**toUpperCase**() + " is wrong!");

    }

**1.9 Comparison operators**

**==** Equal to, **!=** Not equal to, **>** Greater, **<** Lower,

**>=** Greater or equal, **<=** Less or Equal,

**=** is just assignment operator.

* Logical operators: **&&** intersection/AND, **||** union/OR, **!** negation/NOT.

Open = True;

if(!Open){

alert("Closed");

} else {

alert("Open");

}

**1.10 CONTROL Statements**

* If: **if**(condition) **statement;** for block **if**(condition){}
* If...Else:

**if**(condition) **statement;**

**else statement;**

For block

**if**(condition)**{};**

**else {};**

* Else if ladder:

**if**(condition)*statement***;**

**else if**(condition)*statement***;**

**else if**(condition)*statement***;**

**else** *statement***;**

For block

**if**(condition){}**;**

**else if**(condition){}**;**

**else if**(condition){}**;**

**else**{}**;**

* Switch: The switch statement is used to perform *different actions* based on *different conditions*. Use the *switch statement* to select one of *many code blocks* to be *executed*.

**switch**(expression) {

**case** x:

// code block

**break**;

**case** y:

// code block

**break**;

**default**:

// code block

}

switch (new **Date**().**getDay**()) {

    case 0:

      day = "Sunday";

      break;

    case 1:

      day = "Monday";

      break;

    case 2:

       day = "Tuesday";

      break;

    case 3:

      day = "Wednesday";

      break;

    case 4:

      day = "Thursday";

      break;

    case 5:

      day = "Friday";

      break;

    case 6:

      day = "Saturday";

  }