Welcome!

Sit with anyone

Announcements

- **Homework will be posted by tomorrow night**
  It's the JavaScript assignment :)
- Please make use of our Q&A queue if you're shy with asking questions
  http://wdd.io/queue
- **Give us feedback:** https://tinyurl.com/wddfa18-feedback

Week 9

Intro to JavaScript

Debugging

Primitive types

Logic operations

Constants & variables

Functions

We've past the halfway point

HTML

Structure

CSS

Design

JavaScript

Function

Linking HTML and JavaScript file

How will the HTML file know where to find its codin' ?!

Inside the `<head>` tag, add this line:

```
<script type="text/javascript" src="assets/scripts/main.js"></script>
```

*No need to memorize this — I'd just copy and paste or something like that

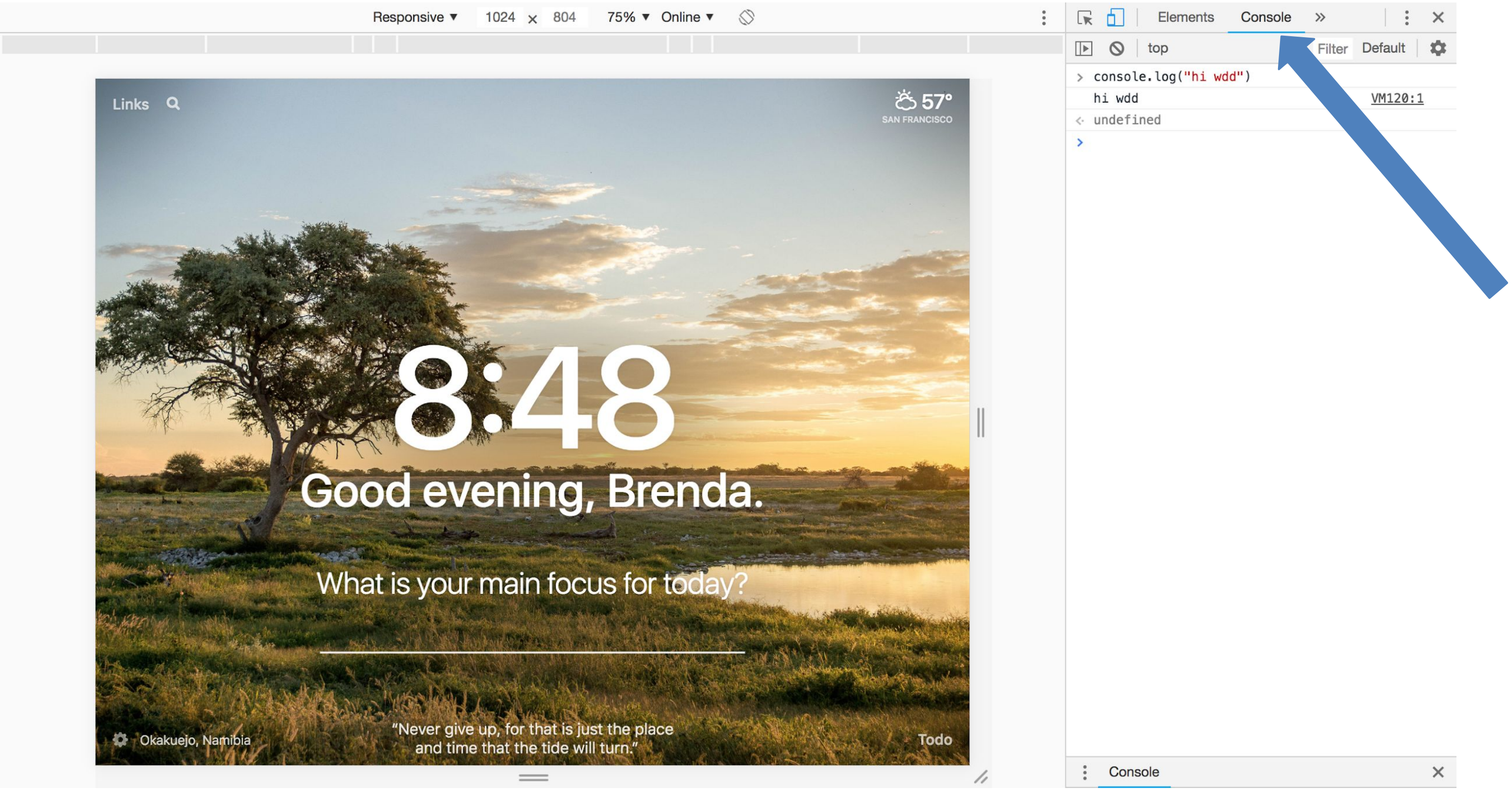
First, debugging

Using the developer tools

- **Always** check the developer tools (e.g. inspect elements) when troubleshooting
  - You should be already doing this
- Use `console.log(“stuff”)`
  - Prints info to the console
  - Put at places where you think the code is broken

Using the developer tools

- **Always** check the developer tools (e.g. inspect elements) when troubleshooting
  - You should be already doing this
- Use **`console.log(“stuff”)`**
  - Prints info to the console
  - Put at places where you think the code is broken



```
> console.log("hi wdd")
  hi wdd                                   VM120:1
<· undefined
>
```

Intro to JavaScript types

A few primitive types

The building blocks of JS (and other programming languages)

- **`number:`** `1, 2, 3, 0.8, 100, 1.6`
- **`string:`** `"hello", "wdd", "wowza cool class"`
- **`boolean:`** `true, false`

Number

Examples:

- `24601` (decimal), `0x3ebd35` (hexadecimal)

Some mathematical operations:

- Regular: `+`, `-`, `*`, `/`
- Power: `*base* ** *exponent*`
- Modulo: `*dividend* % *divisor*` (reasonable with positive numbers)
  `5 % 2 == 1` because `5 / 2 == 2` (floor div), remainder `1`

String

Surrounded by...

- **Either** apostrophes/single-quotes → `'`
- **Or** double-quotes → `"`

A string can be anything!

- `"WDD is great!!"`
- `'123456789'`

`A basic string operation → + (concatenation)`

Boolean

Two possible values: `true`, `false`

Expressions that evaluate to booleans, often times comparisons:

- Loose equality operator → `==`, `!=`
  - `1 + 5 == 6` evaluates to `true`
  - `1 == 2` evaluates to `false`
  - `1 + 5 != 2` evaluates to `true`
- Other relational operators → `>`, `<`, `>=`, `<=`

`typeof` operator

It tells you in a "string" the primitive type of a value

- `typeof 10.24 == "number"`
- `typeof "hello" == "string"`
- `typeof true == "boolean"`

Let's take 4 minutes for a little practice

https://playcode.io/140541?tabs=console&script.js&output

Building on top of booleans

Logic operations

Logic operations

Or → `||`

Just one thing in the expression has to be true

- `true || false`
  evaluates to `true`
- `1 + 1 == 4 || 1 + 1 == 2`
  evaluates to `true`
- `8 == 8 || 1 == 1`
  evaluates to `true`
- `2 == 0 || 1 == 8`
  evaluates to `false`

And → `&&`

Everything in the expression has to be true

- `true && false`
  evaluates to `false`
- `1 + 1 == 4 && 1 + 1 == 2`
  evaluates to `false`
- `8 == 8 && 1 == 1`
  evaluates to `true`
- `2 == 0 && 1 == 8`
  evaluates to `false`

Logic operations (cont'd)

Negation, flipping between `true` and `false` → `!`

- `1 + 1 == 5`

  evaluates to `false`
- `!(1 + 1 == 5)`

  evaluates to `true`
- `!(1 + 1 != 2)`

  evaluates to `true`
- `!!(1 + 1 != 2)`

  evaluates to `false`

JavaScript constants & variables

Always remember…

Every constant & variable must have a unique name

Variables

We can **declare** (once) variables with unique names to hold values for later use

Format:

```
let variableName = value;
```

Example:

```
let numStudents = 120;
```

To reassign a different value: (notice that we don't use `let` here because of reassigning)

```
numStudents = 120;
```

```
// We start with 20 potatoes
let numPotatoes = 20;

// Print it out in the console
console.log("We have", numPotatoes, "potatoes");

// We sold 10 potatoes during the day
// The equal symbol here means assignment
numPotatoes = numPotatoes - 10;

// Print out how much we have left
console.log("We have", numPotatoes, "potatoes");
```

https://playcode.io/140537?tabs=console&script.js&output

Constants

Again, we can **declare** (once) variables with unique names to hold values for later use

Format:

```
const variableName = value;
```

Example:

```
const jacobsFireCode = 140;
```

Once initialized, we **cannot reassign** the variable to a different value :(

```
// A gold potato has a weight of 200 pounds
const goldPotatoWeight = 200;

// Someone comes at night and wanted to change it
:(
goldPotatoWeight = 10;

// TypeError:
// Attempted to assign to readonly property.
```

A first look at functions

Well, the truth is that you've seen at least one function already

`console.log()` is a function

It's a very special one though because it's built-in

And it prints stuff to the console (dev-tool)

Functions

Functions are reusable pieces of code
We can define our own functions too!

In JavaScript, **function is a primitive type** 😮

Syntax:

```
function functionName(arg1, ...) {
   // Do something...
   return returnValue;
}
```

```
function multiply2(number) {
   return number * 2;
}

console.log(1);                       // 1
console.log(multiply2(1));            // 2
console.log(multiply2(multiply2(1))); // 4

console.log(typeof multiply2)         // function

let mul2 = multiply2;

console.log(mul2(4));                 // 8
```

Let's take 4 minutes to practice writing some functions

https://playcode.io/140657?tabs=console&script.js&output

Something new for everybody

Make your panic button

Event handling: A primer

```
<img id="panic-button">


<script type="text/javascript">
  function panicButtonClicked() {
      alert("Ahh! Somebody just clicked the panic button :o")
  }

  // We want the browser to run panicButtonClicked() for us when someone clicks
  // the button with id "panic-button"
  document.getElementById("panic-button").onclick = panicButtonClicked;
</script>
```

https://playcode.io/140602?tabs=script.js&output

Now that we can attach events to elements, what else can JavaScript do?

Here's something pretty amazing...

Drumrolls~

What about setting the styles for elements?

Setting inline style in JavaScript

```
<img id="pumpkin">
<div>Width: <input id="pumpkin-width" type="range" min="10" max="200" value="100"></div>

<script type="text/javascript">
 // Note that document.getElementById("id-of-some-element") gives you an element
 // Then you can use element.style.cssPropertyName = cssPropertyValue to change its style

 function pumpkinSizeUpdated() {
   const width = document.getElementById("pumpkin-width").value;
   const pumpkin = document.getElementById("pumpkin");
   pumpkin.style.width = width + "px";
 }

 // We want the browser to run pumpkinSizeUpdated() when the range slider is changed immediately
 document.getElementById("pumpkin-width").oninput = pumpkinSizeUpdated;
 </script>
```

See demo on next slide...

Let's try out this demo

https://playcode.io/140670?tabs=console&index.html&output

A first special statement

if ... else ...

If-statements

We use if-statements so the code produces different results based on varying conditions

Syntax:

```
if (condition) {
   // When the condition satisfies
} else {
   // Otherwise :(
}
```

```
let temperature = 19;
const bayAreaAvgTemperature = 14;

if (temperature >= bayAreaAvgTemperature) {
 console.log("The temperature's not so cool");
} else {
 console.log("It's getting colder");
}
```

https://playcode.io/140570?tabs=console&script.js&output