GENERAL ASSEMBLY

VARIABLES AND CONDITIONALS

Jonathan Cheng

REVIEW

LEARNING OBJECTIVES

- Practice programmatic thinking by writing pseudo code to solve problems
- Define website behavior and practical uses of JavaScript
- Predict DOM output / changes by reading JS code

LEARNING OBJECTIVES

LEARNING OBJECTIVES

- Work with Javascript variables to store information
- Execute different blocks of code based on certain conditions
- Understand the types of data Javascript understands and how we write them (syntax)
- Perform basic arithmetic with numbers using Javascript
- Use the browser’s built in Javascript console to debug and experiment

VARIABLES

- We can tell our program to remember values for us to use later on
- The action of saving a value to memory is called assignment (or colloquially, setting a variable)
- The entity we use to store the value is called a variable

- The action of getting the value from a variable is called accessing the variable (or getting the variable)
- We will use all the above techniques to store values into variables, and generate new values using existing variables

VARIABLES

Declaration: `var age;`

Assignment: `age = 21;`

Both at the same time: `var age = 21;`

VARIABLES

```
var name = “Jo”;
```

```
name = “Amir”;
```

Note: name is now Amir.

- Variables start with a lower case letter
- If they contain multiple words, subsequent words after the first start with an upper case letter. We call this style of naming camelcase, the most popular naming style in javascript

```
var numberOfStudents = 10;
```

CODEALONG

SCORE KEEPER

DATATYPES

The types of values Javascript supports include:

- Strings: text, strings of characters
- Ints: integers, whole numbers
- Floats: decimals, floating point numbers
- Booleans: true/false

STRINGS

Strings store text as a string of characters surrounded by quotes:

```
“How is the weather today?”
```

```
“Warm”
```

```
“a”
```

Strings can use double quotes or single quotes, but the quote styles must be paired

```
"How is the weather today?”
```

```
“Warm'
```

```
'a'
```

Since quotes signal the start and end of strings, if we want to include them in the string itself, we have to use special characters to tell the computer. This is called escaping the character.

"They \"purchased\" it”

'It\'s a beautiful day'

Escaping quote characters isn't necessary if the character isn't the same quote type as the start and end quote of the string

'They "took care" of him'

"It's a beautiful day"

CONVERTING STRINGS TO NUMBERS

- var intString = “4”;
- var intNumber = parseInt(intString, 10);
- var floatString = “3.14159";
- var floatNumber = parseFloat(floatString);

CONVERTING NUMBERS TO STRINGS

- var number = 4;
- var newNumber = number.toString();

WHY DO WE NEED TO KNOW THAT?

NUMBERS

Numbers represent numerical data. They can be integers(ints) or decimals (floats).
Integers and floats can be positive or negative

int: -42

float: 3.14 (314×10^{-2})

ARITHMETIC

+ Addition: 8 + 4 = 12

- Subtraction: 8 - 4 = 4

* Multiplication: 8 * 4 = 32

/ Division: 8 / 4 = 2

% Modulus: 8 % 4 = 0 (finds the remainder of division)

CONDITIONALS

```
if(age > 18){
  console.log(“You are an adult”);
}
else{
  console.log(“You are still a minor”);
}
```

MAKING COMPARISONS

Operator	Description	Comparing	Returns
==	equal to	x == 8	FALSE
===	exactly equal to(value and type)	x=== “3”	FALSE
		x===3	TRUE
!=	is not equal	x!=8	TRUE
!==	is not equal(neither value nor type)	x!== “3”	TRUE
		x!==3	FALSE
>	greater than	x>8	FALSE
<	less than	x<8	TRUE
>=	greater than or equal to	x>=8	FALSE
<=	less than or equal to	x<=8	TRUE

```
if(age >= 21){
  console.log(“Congrats, drink up!”);
}
else if(age >=18){
  console.log(“Congrats, you can vote!”);
}
else{
  console.log(“Hmm, you’re pretty young!”);
}
```

CODEALONG
COMPARE NUMBERS

MULTIPLE COMPARISONS

Sometimes we'd like to string together multiple conditions:

AND - &&	TRUE	FALSE
TRUE	true	false
FALSE	false	false

```
if(age >= 18 && age < 21){
  //You can vote but can't
  //drink!
}
```

OR - ||	TRUE	FALSE
TRUE	true	true
FALSE	true	false

```
if(day == “Monday” || day == “Wednesday”){
  //We have class today
}
```

CODEALONG

BLACKOUT

LAB
WEATHER APP

- All together, let's spec out a weather app that takes celsius temperature, converts it to Fahrenheit and changes the background to match said temperature.
- In groups, write pseudo code for the application
- In pairs, write the application (you should borrow heavily from the code alongs we've used today and Monday)

WRAP UP

LEARNING OBJECTIVES

- Work with Javascript variables to store information
- Execute different blocks of code based on certain conditions
- Understand the types of data Javascript understands and how we write them (syntax)
- Perform basic arithmetic with numbers using Javascript
- Use the browser’s built in Javascript console to debug and experiment