

JavaScript



ECOM SCHOOL

המכללה למקצועות הדיגיטל וההייטק

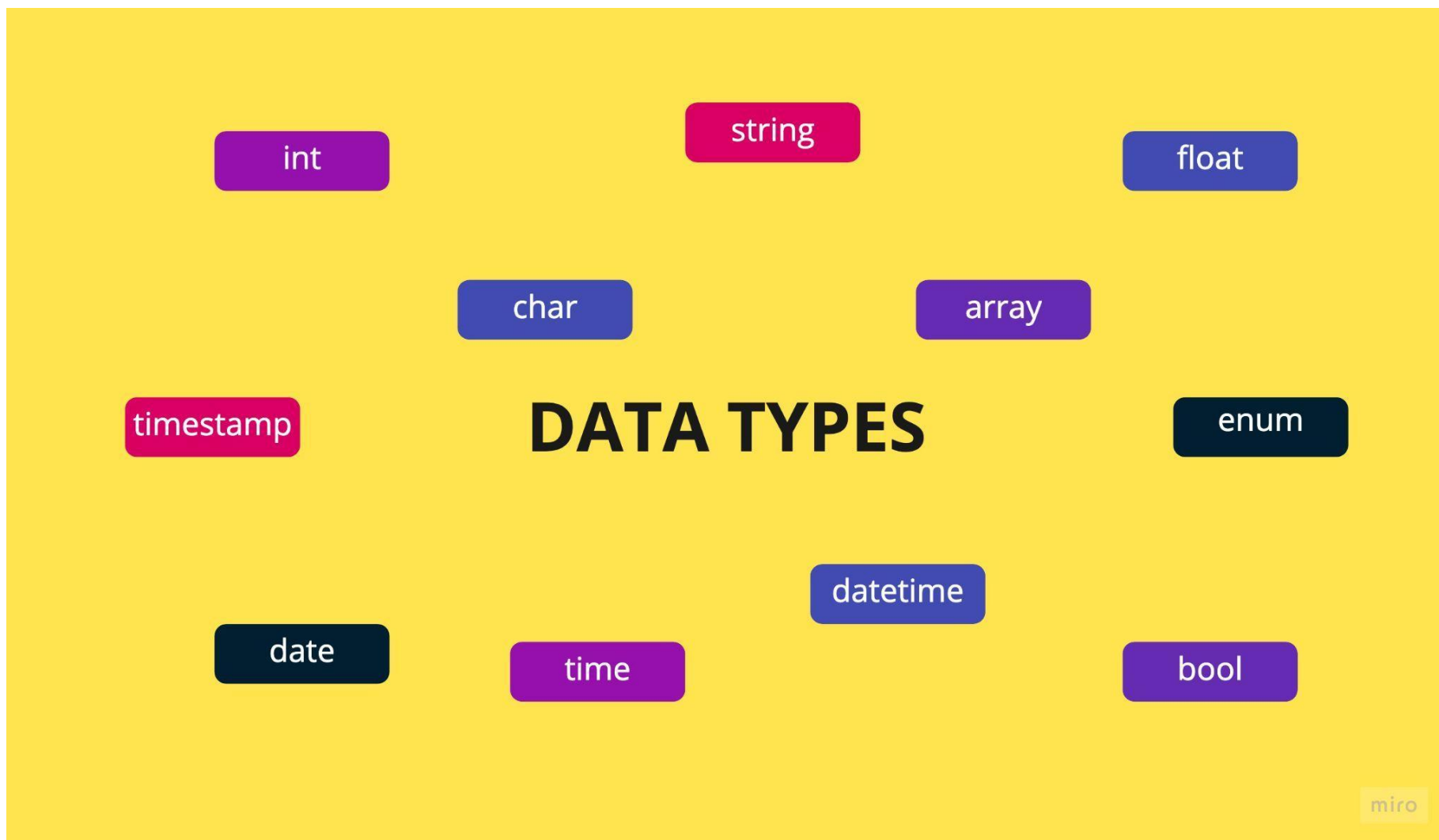
Last lecture reminder



We learned about:

- What is Javascript and what we can build by using it
- How our browser execute javascript code
- Client side Vs Server side and the Client - Server Model
- Javascript features (compiled JIT, dynamic typing, exc...)
- Variables in Javascript (let, var, const)
- Operators in Javascript (Arithmetic operators, Relational operators, exc ...)
- True / False statements by using conditional operators (&& , || , !)

Data Types in Javascript



Javascript Data Types

- **Numbers** - Use for calculations and perform mathematical actions

example.js

```
let numberOfClasses = 10;  
let maxStudentsInClass = 35;  
  
var totalStudents = numberOfClasses * maxStudentsInClass;  
  
var result = totalStudents - 3 * 5 + (6 - 4 );  
  
console.log("This is the result" + result);
```

Output:

337



Javascript Data Types

- **Strings** - Use to represent text, strings are combinations of chars
In Javascript everything that is under double quotes (") is define as a single string
We can concatenate strings by using the + operator

example.js

```
var emptyString = ""  
var notEmptyString = "  
var studentFirstName = "Ben";  
var studentLastName = "Meir";  
  
console.log("This is the student full name: " + studentFirstName + " " + studentLastName);
```

Output:

This is the student full name: Ben Meir

Javascript Data Types

- **Boolean** - This is a special variable that can hold only two values: "true" or "false"

We will use boolean data type when we want to create a "flag" that according to its value we will do an action

```
17 |
18 | const maxNumberInClass = 10;
19 | const minimumAge = 18;
20 | var registerCount = 0;
21 | var isAllowedToRegister = true;
22 |
23 | function canYouRegister (studentName, age) {
24 |     if (age > minimumAge && isAllowedToRegister){
25 |         registerCount++;
26 |         isAllowedToRegister = (registerCount < maxNumberInClass);
27 |         console.log(studentName + " Successfully Registered!");
28 |     } else {
29 |         console.log(studentName + " Could not registered");
30 |     }
31 | }
32 |
33 | canYouRegister("ben", 19);
34 | canYouRegister("noam", 25);
```

Output:

```
ben Successfully Registered!
noam Successfully Registered!
```

Javascript Data Types

- **Null & undefined** - null and undefined are special data types that responsible to represent empty. when a variable is holding an empty value, Javascript will represent this value as “null”
When we created a variable without assigning to it any value, Javascript will represent it as “undefined”

```
8 var nullVariable = null;  
9 var undefinedVariable;  
0  
1 console.log("This is null variable: " + nullVariable);  
2 console.log("This is undefined variable: " + undefinedVariable);
```

Output:

```
This is null variable: null  
This is undefined variable: undefined
```

Javascript Data Types

- **Null & undefined** - When we will try to do programming actions on null variables in some cases they will not affect our functions and in other cases our program will return “undefined”

For example:

```
6
7 var nullVariable = null;
8 var someNumber = 10;
9
10 function someFunction() {
11   if (nullVariable > someNumber) {
12     return ("True");
13   } else {
14     return ("False");
15   }
16 }
17
18 console.log(nullVariable + 5);
19 console.log(someFunction());
20
```

Output:

5

False

Javascript Data Types

- Returning undefined as response:

```
7  
8 var someArray = ["Ben", "Noam", "Sigal"];  
9  
10 console.log(someArray[1])  
11 console.log(someArray[5])  
12  
13  
14
```

Output:

Noam

undefined



JS Functions



What is a programming function?

Programming functions in general giving you the ability to define code that your program can execute any time and anywhere in your code

By using function we can:

- Use the same block of code everywhere we want without rewrite it
- Run code only when something that we want happen (like pressing a button)
- Keep our code short and clean
- Better explain our code to other developers

Functions in general represent this flow:



Functions in Javascript

By using the **function** Javascript keyword we are declaring on a new function

In order to create a function in Javascript we have to use the function keyword

```
7 |  
8 | function myFirstFunction() {  
9 |   console.log("Hey! This is my first function!")  
10 | }  
11 |  
12 | myFirstFunction()
```

Output:

Hey! This is my first function!

After declaring a function we need to give it a **name**

Same as with variables, Javascript allow us to choose any name you want

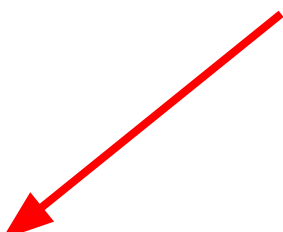
We should choose names to our functions that represent the purpose of the function

For example: validateStudentAge() , getName(), exc...

Functions in Javascript

After we determined the name of our function we need to choose the **input** our function should get in order to produce the output, We call it **Parameters**

- We should put the function parameters between - ()
- We can pass as much parameters as we want to a single functions - (parameter1, parameter2, parameter3, ...)
- If our function has no parameters we will use empty ()
- Without providing all the parameters the function could not be executed and we will get an error



```
6  
7 function myFirstFunction(name){  
8   return "Hey! This is " + name + " first function!!!!!"  
9 }  
10  
11 console.log(myFirstFunction("Ben"))
```

Output:

Hey! This is Ben first function!!!!



ECOM SCHOOL


המכללה למקצועות הדיגיטל וההייטק

Functions in Javascript

The “process” part of the function is what we put between {}

Everything between {} is been called **function scope** or **function block**

- Every code inside the function scope is been executed line by line when we call it, everytime we call it
- Inside the function scope we can use all the parameters this function got by calling their names
- In addition, We can create new variables that will only be exists inside the function scope



```
6  
7 function myFirstFunction(name){  
8   return "Hey! This is " + name + " first function!!!!!"  
9 }  
10  
11 console.log(myFirstFunction("Ben"))
```

Output:


Hey! This is Ben first function!!!!!

Functions in Javascript

The “output” part of the function is what we **return** when the function finish to execute all the code inside the function scope

By using the **return** keyword we determine what is the output of our function

- Return should only been put at the end of the function
- The return keyword end the function execution and “close” it
- We can put more than one return inside a function but its best practice to avoid that and use just one return
- It's not mandatory to return a value, so we can create functions that are not returning anything and doesn't use the return keyword



```
6  
7 function myFirstFunction(name){  
8   return "Hey! This is " + name + " first function!!!!!"  
9 }  
10  
11 console.log(myFirstFunction("Ben"))
```

Output:

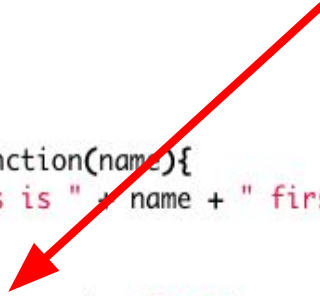
```
Hey! This is Ben first function!!!!!
```

Functions in Javascript

How to call Javascript function?

- Make sure that the function you want to call is already been created
- In the code outside the function scope write the function name and put () after it
- If the function required parameters put them inside the () and pass them by using , between them
- If the function return a value you can save it to a variable or use it in any way you want to

```
7 function myFirstFunction(name){  
8   return "Hey! This is " + name + " first function!!!!!"  
9 }  
10  
11 console.log(myFirstFunction("Ben"))
```



Output:

```
Hey! This is Ben first function!!!!!
```


Class Exercise - Functions



Class Exercise - Functions



Instructions:

Go to <https://onecompiler.com/javascript>

Your mission is to create a basic calculator with all 4 basic arithmetic actions

Your calculator should only work with two numbers at each time

Your calculator should implement the following:

- Add function ($8 + 4 = 12$)
- Reduction function ($8 - 4 = 4$)
- Multiple function ($8 * 4 = 32$)
- Division function ($8 / 4 = 2$)

You should test your calculator by calling each function and print the output to the console



Class Exercise - Functions Solution

