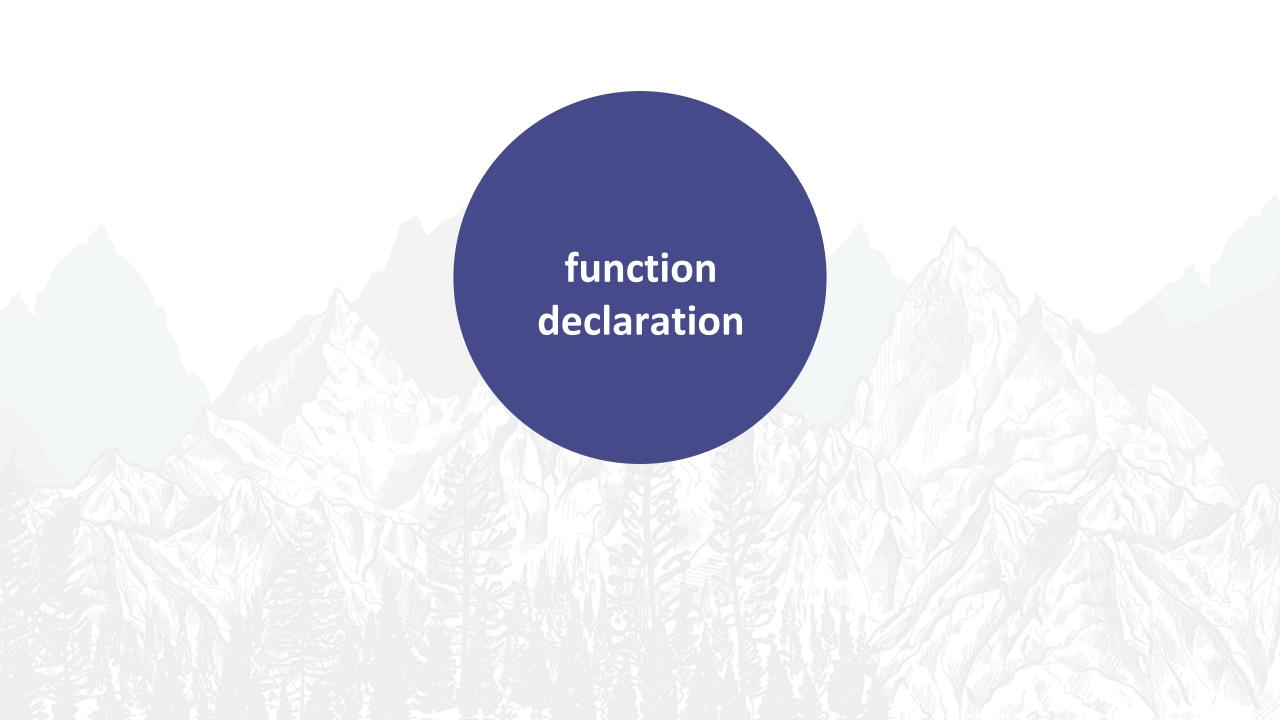# functions

PHP web development 2020/2021

Milena Tomova

Vratsa Software

https://vratsasoftware.com/

Table of Contents

1. function declaration/definition/satement
2. function expression
3. IIFE

function declaration

also function definition, function statement

```
function name(parameter1, parameter2, parameter3) {
  // code to be executed
}
```

Function **parameters** are listed inside the parentheses () in the function definition.

Function **arguments** are the **values** received by the function when it is invoked.

function declaration

The code inside the function will execute when "something" **invokes** (calls) the function:

- When an event occurs (when a user clicks a button)
- When it is invoked (called) from JavaScript code
- Automatically (self invoked)

function declaration

invoking a function - example 1

```
name(arg1, arg2, arg3);
```

function declaration

example 2.a

```
function toCelsius(fahrenheit) {
  return (5/9) * (fahrenheit-32);
}
```

A	`var x = toCelsius(77),` `    text = "The temperature is " + x + " Celsius";`

B	`var text = "The temperature is " + toCelsius(77) + " Celsius";`

function expression

The main difference between a function expression and a function declaration is the *function name*, which can be omitted in function expressions to create *anonymous* functions.

```javascript
const getRectArea = function(width, height) {
  return width * height;
};


console.log(getRectArea(3, 4));
// expected output: 12
```

function declaration

- A function expression can be used as an IIFE (Immediately Invoked Function Expression) which runs as soon as it is defined. See also the chapter about functions for more information.

IIFE

IIFE

Immediately Invoked Function Exppressions
        - a shielded scope in the global scope

- a function that dies immediately after it came to life - after it is invoked
- executes immediately the script is loaded
- global scope is not poluted with variables and functions
- the variables and functions declared inside of the IIFE are shielded - cannot be reached from the outside of the IIFE scope accidentally or intentionally

IIFE

```
(function () {

    //statements - variables, functions declared and
invoked inside the IIFE

})();
```

IIFE

```
(function () {

    //statements - variables, functions declared and
invoked inside the IIFE

})()
```

anonymous function with lexical scope enclosed within the Grouping Operator ()

IIFE

```
(function () {

    //statements - variables, functions declared and
invoked inside the IIFE

})()
```

creates the immediately invoked function expression () through which the JavaScript engine will directly interpret the function.

IIFE - examples

```javascript
(function () {
    var aName = "Barry";
})();


console.log( aName );


// Variable aName is not accessible from the outside scope
aName // throws "Uncaught ReferenceError: aName is not
defined"
```

IIFE - examples

```
var result = (function () {
    var name = "Barry";
    return name;
})();


// Immediately creates the output:

result; // "Barry"
```

Don't try to understand
JavaScript
~~girl~~ completely. If you do,
then either you'll go mad..
it
or you'll start loving ~~her~~.

- Unknown

Questions?

Partners

Trainings @ Vratsa Software

- Vratsa Software – High-Quality Education, Profession and Jobs

    - www.vratsasoftware.com

- The Nest Coworking

    - www.nest.bg

- Vratsa Software @ Facebook

    - www.fb.com/VratsaSoftware

- Slack Channel

    - www.vso.slack.com