**Function Advanced**

1. Execution context (*this*) – the function context is the object that owns the currently executed code. Depends on how the function is invoked, the context can be either the global scope – **func()**, object scope – **object.func()** or DOM event – **addEventListener(event, func)**

Methods

- call()  
- apply() -> give the function context as an argument and invoke it **functionName.apply(refferenceToContextObject, [params])**  
- bind() 🡪 binds a context object to the function without calling it

\* The context of an arrow function always retains the reference to the object where it is defined. Call, apply and bind don’t work on arrow functions.

1. Functional programming in JS

**First-class functions** 🡪 treated like a variable, can be passed as an argument, returned by another function, assigned as a value to a variable.

*“The term ‘first-class` means that something is just a value. A first-class function is one that can go anywhere that any other value can go – there few to no restrictions.”*

**High-order functions** 🡪 Take other functions as an argument or return a function as a result

**Predicates** 🡪 any function that returns a bool based on evaluation of the truth of an assertion. Often found in the form of callbacks.

**Build-in Higher Order Functions** :

* Array.prototype.map()
* Array.prototype.filter()
* Array.prototype.reduce()
* Array.prototype.some()
* Array.prototype.every()

**Pure functions** 🡪 Functions that do not have side effects. Returns the same result given same parameters. Execution is independent of the state of the system.

**Referential Transparency** 🡪 An expression that can be replaced with its corresponding value without changing the program’s behavior. Expression is pure and its evaluation must have no side effects.

1. Closure – Inner function state. The scope of an inner function includes the scope of the outer function! The inner function retains variables being used from the outer function scope even after the parent function has returned!
2. Function decoration

* partial application (set arguments without invoking)
* currying – function decomposision