Functions, Hoisting and Scoping

LESSON OBJECTIVES

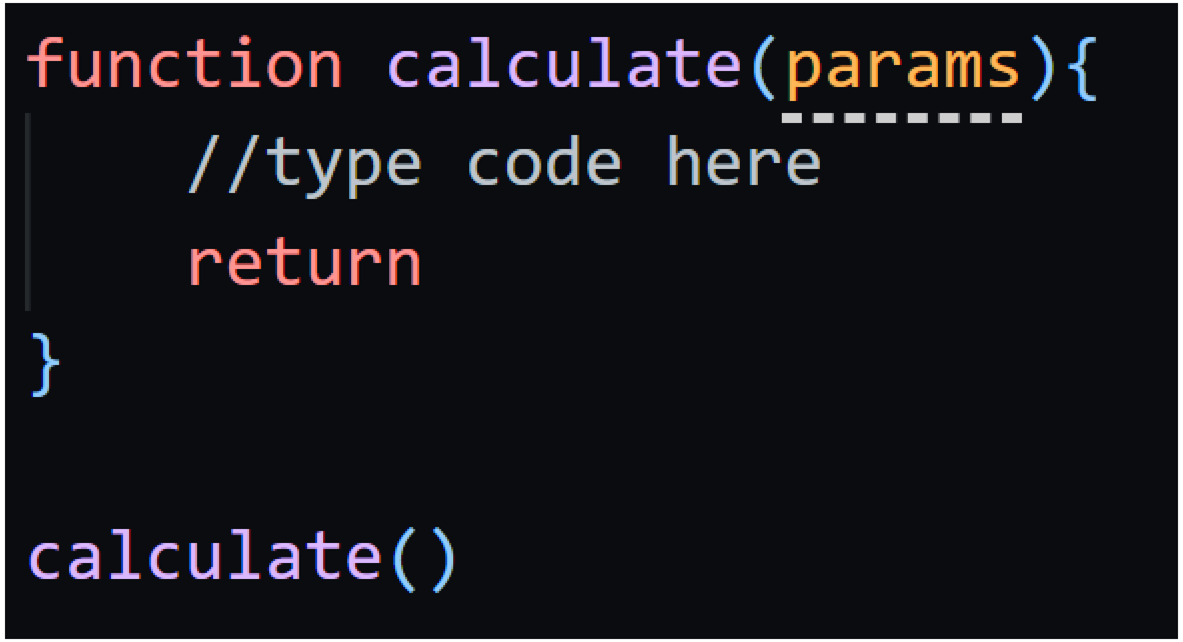
- Understand the syntax of functions
- Create and use functions

Functions

- A section of reusable code
- We can declare the code once, use it wherever and whenever we want
- Call the function to execute the code.
- Functions also allow us to run code at specific instances (event listeners)

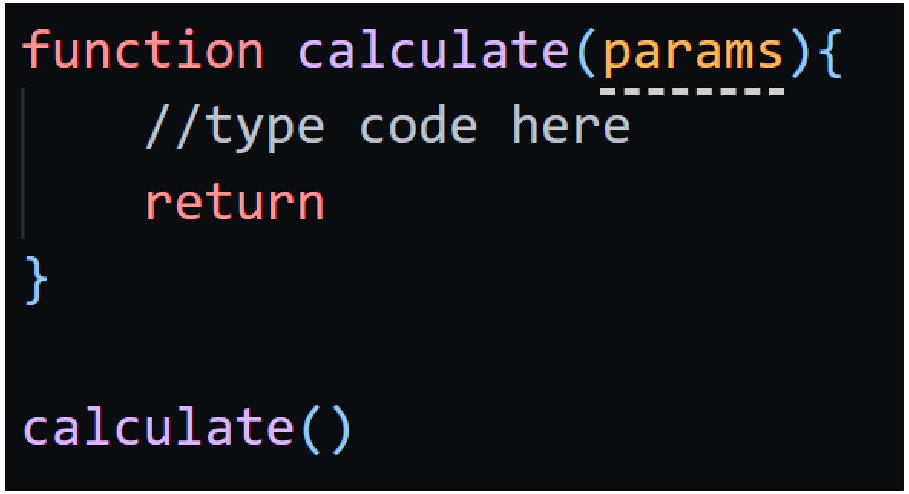
Functions ༼ つ ◕_◕ ༽つ

- Function declaration
- Calling a Function(Execution)

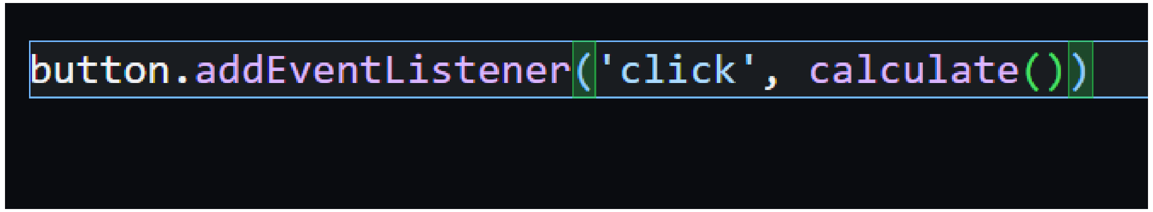


```
function calculate(params){
    //type code here
    return
}

calculate()
```

Functions ~~breakdown



```
function calculate(params){
    //type code here
    return
}

calculate()
```

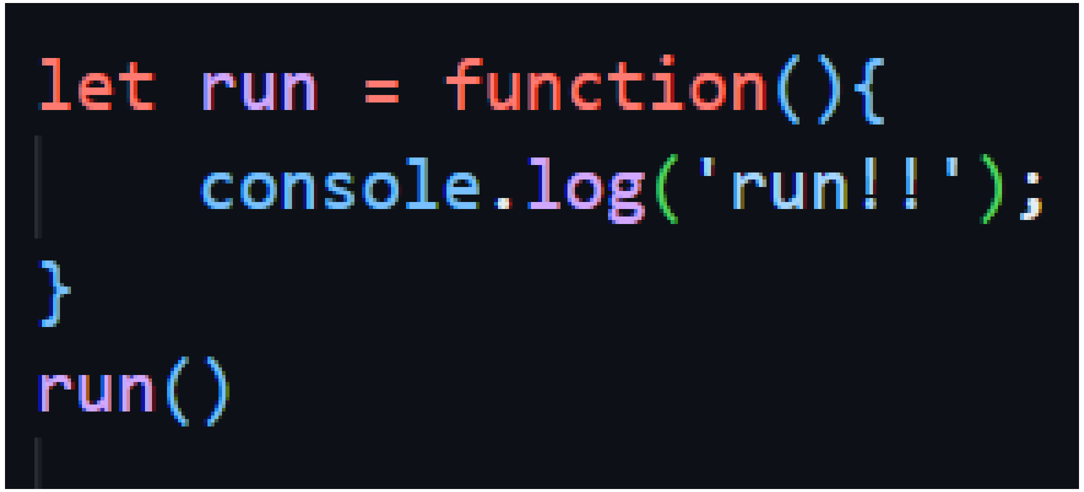


```
button.addEventListener('click', calculate())
```

- Function Usage
- Parameters
- Event Listeners
- ‘Return’ Keyword

Function Expression

- Function Expression(anonymous)
- Calling a Function(Execution)



```
let run = function(){
    console.log('run!!');
}
run()
```

Hoisting

- Hoisting is the process of moving function declarations to the top of the file.
- This is done automatically by the JavaScript engine that is executing this code.
- That's the reason we can call functions that are defined using the function declaration syntax before their definition.

Scoping

- The scope of a variable or a constant, determines where that variable or constant is accessible within the block in which they are defined.
- 2 types: Local and Global scoping
- If statements, Loops, Functions - have a local scope
- Let and Const(Locally scoped) vs Var(Global Scoped)