


Episode 13: First Class Functions → ft. Anonymous Functions

? Q1: What is a Function Statement?

☒ A **Function Statement** is a standard way to define a function.


```
function greet() {
  console.log("Hello!");
}
greet(); // Hello!
```

 **Also known as:** Function Declaration

? Q2: What is a Function Expression?

☒ A **Function Expression** is when a function is assigned to a variable. Here, the function behaves like a **value**.


```
var greet = function () {
  console.log("Hello!");
};
greet(); // Hello!
```

 Think of it like:

```
var number = 5;
var greet = function() {}; // assigning a value (function) to variable
```

Q3: Function Statement vs Function Expression

Feature	Function Statement	Function Expression
Name Required	<input checked="" type="checkbox"/> Yes	<input checked="" type="checkbox"/> Not required (can be anonymous)
Hoisting Behavior	<input checked="" type="checkbox"/> Hoisted fully	<input checked="" type="checkbox"/> Not hoisted (variable is hoisted as undefined)
When Usable	Before and after declaration	<input checked="" type="checkbox"/> Only after the line is parsed

 Code Example:

```
a(); // Hello A
b(); // ✗ TypeError: b is not a function

function a() {
  console.log("Hello A");
}

var b = function () {
  console.log("Hello B");
};
```

🤔 Why this happens?

- During memory creation phase:
 - `a` is fully hoisted with function definition
 - `b` is hoisted as `undefined`

? Q4: What is a Function Declaration?

- ☒ Another name for a **Function Statement**.

```
function sayHi() {
  console.log("Hi!");
}
```

? Q5: What is an Anonymous Function?

- ☒ A function **without a name**.

```
function () {
  console.log("Hi");
}
// ✗ SyntaxError - function statements need names
```

🔗 Use Case:

- ☒ Anonymous functions are typically used in **function expressions** or **callbacks**.

```
var greet = function () {
  console.log("Hello!");
};
```

? Q6: What is a Named Function Expression?

- ☑ Like a function expression, but the function has a name.

```
var greet = function welcome() {  
  console.log("Hello from welcome!");  
};  
  
greet(); // ☑ Works  
welcome(); // ✗ ReferenceError: welcome is not defined
```

🗯 **Note:** The name `welcome` is only accessible inside the function body.

? Q7: Parameters vs Arguments?

Term	Meaning
Parameter	Variable name in function definition
Argument	Actual value passed during function call

```
function sum(a, b) { // a, b = parameters  
  return a + b;  
}  
sum(3, 5); // 3, 5 = arguments
```

🗯 Q8: What are First Class Functions (aka First-Class Citizens)?

- ☑ In JavaScript, functions are **First Class Citizens**.

This means:

- You can **pass them as arguments**
- You can **return them from other functions**
- You can **assign them to variables**

```
var printFn = function (param1) {  
  console.log(param1);  
};  
  
printFn(function () {  
  console.log("I'm a function!");  
});
```

🔗 Returning a function:

```
function outer() {  
  return function inner() {  
    console.log("Returned function");  
  };  
}  
  
const result = outer();  
result(); // Returned function
```

🧠 This enables patterns like **Higher Order Functions**.
