Function declarations vs expressions

Let's explore **Function Declarations vs Function Expressions** in JavaScript — two fundamental ways to define functions.

🔷 1. Function Declaration

✅ Syntax:

```
function functionName(parameters) {
  // function body
}
```

✅ Example:

```
function greet(name) {
  return `Hello, ${name}!`;
}

console.log(greet("Abhi")); // Output: Hello, Abhi!
```

✅ Characteristics:

- **Named function**: Has a defined name.
- **Hoisted**: Can be called **before** the function definition in the code (thanks to hoisting).
- Available in the **entire scope** where it's defined.

🧠 Hoisting Example:

```
sayHi(); // ✅ Works fine

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

🔷 2. Function Expression

✅ Syntax:

```
const functionName = function(parameters) {
  // function body
};
```

✅ Example:

```
const greet = function(name) {
  return `Hello, ${name}!`;
};

console.log(greet("Abhi")); // Output: Hello, Abhi!
```

✅ Characteristics:

- Can be **anonymous** (without a name).
- Can also be **named expressions**, but the name is only accessible inside the function itself.
- **Not hoisted**: Cannot be called before its definition.
- Commonly used for **inline functions**, **callbacks**, and **closures**.

❌ Hoisting Example:

```
sayHi(); // ❌ ReferenceError: Cannot access 'sayHi' before initialization

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

🆚 Summary: Function Declaration vs Expression

Feature	Function Declaration	Function Expression
Syntax	`function name() {}`	`const name = function() {}`
Hoisted	✅ Yes	❌ No
Named	Always named	Can be named or anonymous
Called before define?	✅ Yes	❌ No
Common uses	General-purpose functions	Callbacks, closures, immediate invocation

👇 Bonus: Use in Callbacks

```
setTimeout(function() {
  console.log("This is a function expression!");
}, 1000);
```

✅ When to Use

Use Case	Recommended Style
Reusable utility/helper function	Function Declaration
Inline or anonymous logic	Function Expression
Closures or dynamic behavior	Function Expression

Would you like to follow up with **arrow functions** next?