/academeez

Variable Declaration

`var, let, const`

Variable declaration

- During our Javascript app, we will use variables to store information.
- In this lesson we will learn about different ways to create variables in Javascript

var

- Declares a variable
- The variable scope will be in the function it is in
- If we are outside the function the variable will be global on the browser, and local to the module on node
- var varname1 [= value1] [, varname2 [= value2] ... [, varnameN [= valueN]]];

```
var a = 10;

if (someTrueBoolean) {
    var a = 20;
}

// there is only one variable a regardless of line 20
console.log(a);
```

var quirks

- Variables can be referenced before they are declared - hoisting
- Variables with no var are declared as globals in non strict mode
- Can declare multiple variables in a single line

```
// Hoisting
console.log(a); // undefined
var a = 10

b = 20; // might be a global variable if we are not in strict mode
```

var bug

academeez

- The function scope can, in some cases cause bugs

```
function messages() {
    var message = 'this is a message printed in function execution';

    for (var i=0; i<1000; i++) {
        var message = 'I want an internal loop message';
        console.log(message);
    }

    console.log(message);
}
```

let

- Declares a variable
- The variable scope will be inside the block
- If we are outside a block it will be available in the current module
- No hoisting
- `let var1 [= value1] [, var2 [= value2]] [, ..., varN [= valueN];`

let example

academeez

```
// no hoisting
console.log(hello);
let hello = 'world';


function messages() {
    let message = 'this is a message printed in function execution';

    for (var i=0; i<3; i++) {
        let message = 'I want an internal loop message';
        console.log(message);
    }

    console.log(message);
}

messages();
```

const

- Declares a variable
- The variable has to be assigned on declaration
- The variable scope is in the block
- const name1 = value1 [, name2 = value2 [, ... [, nameN = valueN]]];

```
function greetings(name) {
    return (req, res) => {
        res.send(`Hello ${name}`);
    }
}

app.get('/piglet', greetings('Piglet'));
app.get('/sweetness', greetings('Sweetness'));
```

const - examples

academeez

```
// error -  has to be assigned
const hello;
hello = 'world'

// error - cannot be reassigned
const foo = 'bar';
foo = 'foo';

// error - scope in block
if (foo === 'bar') {
    const message = 'hello world';
}

console.log(message);

// const can be muted
const myArray = [];
myArray.push('hello');
```

const/let/var - main differences

academeez

	Scope	Number of assignments
var	Function	∞
let	{}	∞
const	{}	1

const/let/var - which to choose

- Start with the most strict - **const**
- If the variable needs more than one assignment than move to **let**
- If the variable needs to be available outside of the block scope use **var**
- It is common to enforce these regulations with linting tools

Summary

- Prior to ES6 Javascript had one way to declare variables, using the **var** keyword
- Now we can also declare variables using **const / let**
- We now have to consider where our variable will live and how many assignments we will have to the variable
- Using the different variable declarations properly will avoid variable collision bugs

academeez

Thank You

Next Lesson: If