ES6

ES6

ES2015

- Javascript의 문법을 정리 한 것.
- 기존 Javascript의 불편한 점들을 개선하고 추가 기능들을 제공하는 버전.
- ES2020까지 정의되어 있으며 최신 버전을 브라우저나 엔진들이 제공하는 것은 아니다.
- https://kangax.github.io/compat-table/es6/

변수 선언

- let: 주어진 범위 안에서 유효한 변수를 선언 하는 방법
  - var: 기존 변수 선언 방법으로 범위가 함수라 사용하기 어려움
- const: 상수를 선언. 단, reference가 변경되지 않는다면 멤버 변수 등은 수정 가능
- 기존 var 변수: 유효 범위가 함수 단위

- https://es6console.com

변수 선언

- var 방법: var 의 범위는함수.

```
var a=10;

function test(){
  var a=0;
  console.log(a);
  for(var a = 5; a < 7; a++){
    console.log(a);
  }
  console.log(a);
}
console.log(a);
test();
console.log(a);
```

변수 선언

- let 방법

```
let a=10;

function test(){
  let a=0;
  console.log(a);
  for(let a = 5; a < 7; a++){
    console.log(a);
  }
  console.log(a);
}
console.log(a);
test();
console.log(a);
```

람다식

- => 화살표를 이용해 람다식 표기 가능
- 람다식의 경우 부모의 this에 접근할 수 있고(자신의 this를 생성하지 않는다)
- 람다식이 파라미터로 전달 될 경우 다른 파라미터를 참조할 수 있음
- 객체의 생성자, 객체의 함수를 선언할 때는 사용하면 안된다

람다식

this

```
class SayHello{
  constructor(name){
    this.name=name
  }

  funcTypeHello(){
    const handler = function(){
      return `Hello, ${this.name}`;
    }
    return handler();
  }

  lambdaTypeHello(){
    const handler = ()=>{
      return `Hello, ${this.name}`;
    }
    return handler();
  }
}

hello = new SayHello('user')
console.log(hello.funcTypeHello());
console.log(hello.lambdaTypeHello());
```

Object

- 좀 더 편리한 방법으로 객체를 정의할 수 있다.
- 기존

```
const obj={
  name:'user',
  getName: function(){
    return this.name;
  }
};

console.log(obj.getName());
```

신규 (속성으로 외부 변수 사용 가능, 함수 표기법 축약)

```
const name='user';

const obj={
  name,
  getName(){
    return this.name;
  }
};

console.log(obj.getName());
```

Template literal

- ` 를 이용해 탬플릿 문자열을 정의한다. 줄바꿈을 표현하기 쉬우며 변수를 이용한 문자열 합성이 쉽다.

- 기존

```
var body='<body>\n'+
    '<h1>Welcome</h1>\n' +
    '</body>';
var name='abc';
var message = 'Hello ' + name;

console.log(body);
console.log(message);
```

신규

```
var body=`<body>
    <h1>Welcome</h1>
</body>`;
var name='abc';
var message = `Hello ${name}`;

console.log(body);
console.log(message);
```

Modules

- export 변수, 함수로 모듈 생성
  - export default: 이 모듈에 객체가 하나만 있다는 뜻. 불러올때 {} 없이 가능. 불러올 때 이름을 자유롭게 지을 수 있음.
  - import { 변수 또는 함수 } from ‘파일 경로’ 로 모듈 불러오기
- exports.xxx=… (exports == module.exports)
  - exports 객체에 값을 추가하여 전달.
  - require(‘module’)로 불러온다