

# Web Studio 2019

## 8. javascript

김상근 / [sisobus1@gmail.com](mailto:sisobus1@gmail.com)

# Contents

1. 수업을 위한 Layout 제작
2. javascript?
3. javascript
4. javascript!

# Contents

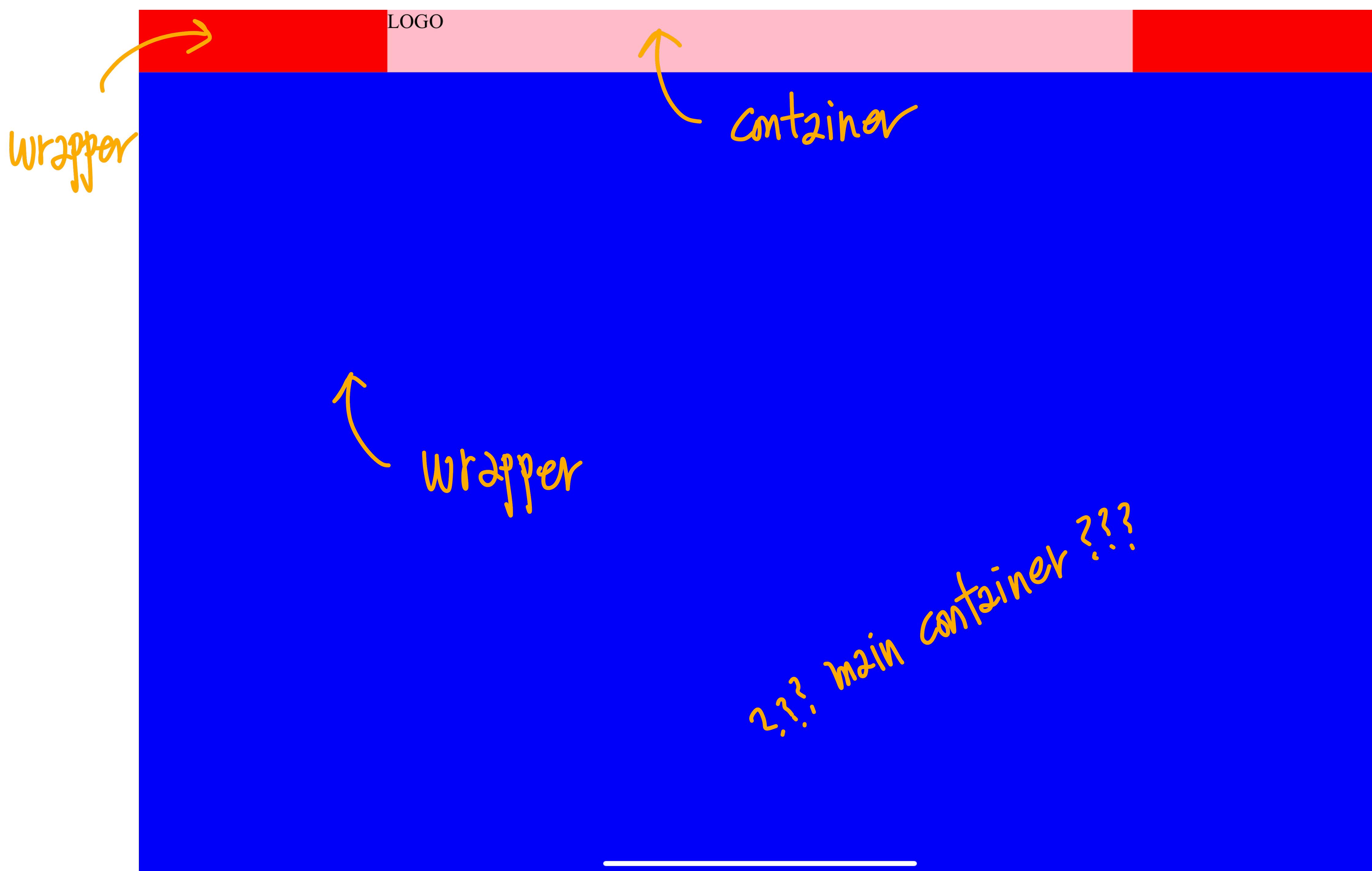
1. 수업을 위한 Layout 제작
2. javascript 란 뭘까? 왜쓸까? 언제쓸까?
3. javascript 에 대해 알아보자
4. javascript 를 적용해보자

# 실습을 위한 Layout 제작

## 1. 대충 틀을 잡아줌, 색도 대충대충

```
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>test</title>
    <link rel="stylesheet" type="text/css" href="./style.css">
  </head>
  <body>
    <div class="navbar-wrapper">
      <div class="navbar-container">
        <p>LOGO</p>
      </div>
    </div>
    <div class="main-wrapper">
      <div class="main-container">
      </div>
    </div>
  </body>
</html>
```

```
html, body {
  padding: 0;
  margin: 0;
  width: 100%;
  height: 100%;
}
p {
  margin: 0;
}
.navbar-wrapper {
  height: 50px;
  background: red;
}
.navbar-container {
  width: 60%;
  height: 50px;
  margin: 0 auto;
  background: pink;
}
.main-wrapper {
  width: 100%;
  height: 100%;
  background: blue;
}
.main-container {
  width: 70%;
  background: green;
}
```



# 실습을 위한 Layout 제작

## 2. 글자도 대충 넣어보고

```
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>test</title>
    <link rel="stylesheet" type="text/css" href=".//style.css">
  </head>
  <body>
    <div class="navbar-wrapper">
      <div class="navbar-container">
        <p>LOGO</p>
      </div>
    </div>
    <div class="main-wrapper">
      <div class="main-container">
        asjkdjfajklsdhfklijasdfhaksdfasdfkljashfdkasdfh<br>
        asldkfjhaskldfhaskldfjasdl
      </div>
    </div>
  </body>
</html>
```



# 실습을 위한 Layout 제작

## 3. Article 샘플도 대충대충

```
<div class="main-wrapper">
  <div class="main-container">
    <div class="article-container">
      <div class="article">
        <h1 class="title">
          updated title
        </h1>
        <p class="author">
          Sangkeun Kim
        </p>
        <p class="content">
          Lorem ipsum dolor sit amet, consectetur adipiscing elit. 
          rus. Mauris at magna placerat, porttitor augue at, dictum sem. 
          Do
          psum, euismod a turpis id, imperdiet luctus elit. Donec eleifend
          esent at porttitor erat. Nulla facilisi. Aenean et hendrerit nisi
          es scelerisque ipsum, vestibulum cursus sem viverra sit amet. Pra
          uat.
        </p>
      </div>
    </div>
  </div>
</div>
```

```
#logo {
  font-size: 20pt;
  padding-top: 10px;
}

.main-wrapper {
  width: 100%;
  height: 100%;
  background: blue;
}

.main-container {
  width: 70%;
  margin: 0 auto;
  background: green;
}

.article-container {
  width: 100%;
  padding-top: 20px;
}

.article {
  border: 1px solid gray;
  padding: 10px;
  width: 100%;
  background: #987654;
}

.article .title {
}

.article .author {
  color: gray;
  font-size: 10pt;
}

.article .content {
  margin-top: 10px;
}
```

# updated title

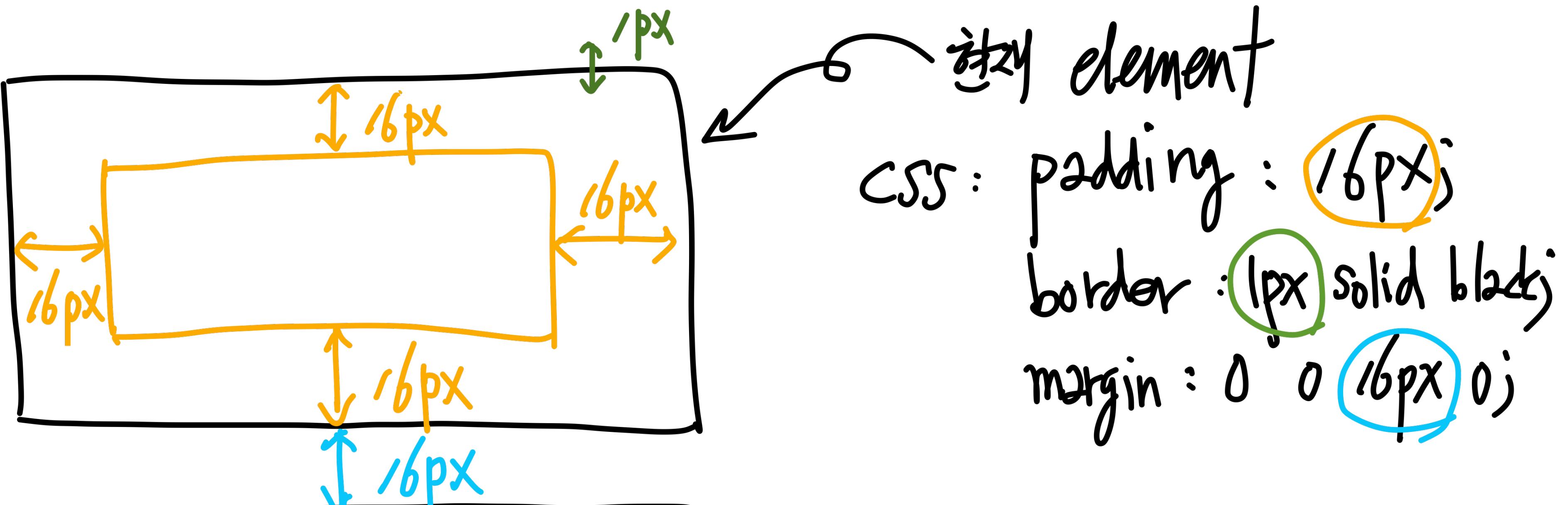
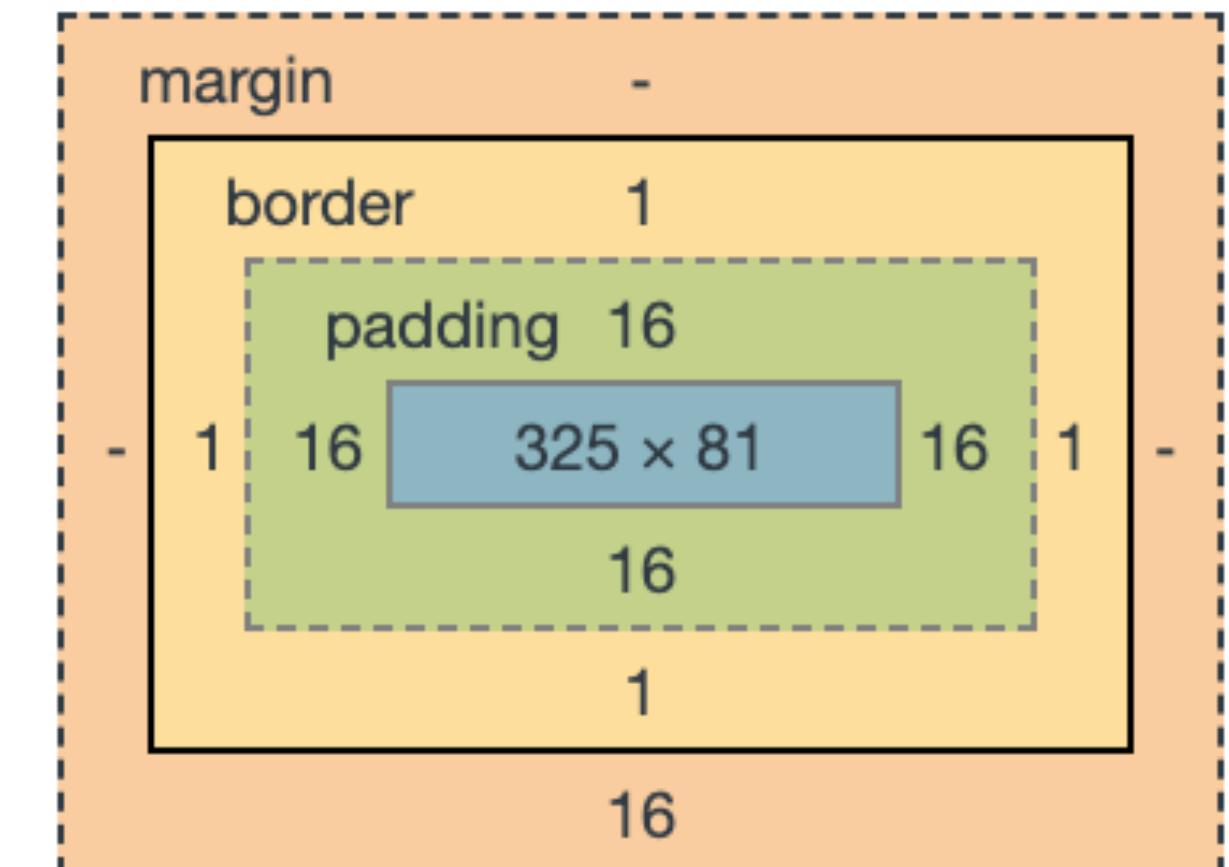
Sangkeun Kim

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed eu ipsum elit. Sed at fermentum purus. Mauris at magna placerat, porttitor augue at, dictum sem. Donec sodales neque nisl. Aenean ipsum ipsum, euismod a turpis id, imperdiet luctus elit. Donec eleifend lacus vel eros sagittis facilisis. Praesent at porttitor erat. Nulla facilisi. Aenean et hendrerit nisi, ac egestas elit. Suspendisse ultricies scelerisque ipsum, vestibulum cursus sem viverra sit amet. Praesent dapibus eleifend nisl non consequat.

# 실습을 위한 Layout 제작

## margin, padding

1. margin은 다른 HTML element들과의 여백
2. padding은 자식 HTML element들과의 여백
3. 한번에 쓸 때는 위, 오른쪽, 아래, 왼쪽 순서



```
<div class="article-container">
  <div class="article">
    <h1 class="title">
      updated title [1]
    </h1>
    <p class="author">
      Sangkeun Kim
    </p>
    <p class="content">
      Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed eu ipsum elit. Sed at fermentum purus. Mauris at magna placerat, porttitor augue at, dictum sem. Donec sodales neque nisl. Aenean ipsum ipsum, euismod a turpis id, imperdiet luctus elit. Donec eleifend lacus vel eros sagittis facilisis. Praesent at porttitor erat. Nulla facilisi. Aenean et hendrerit nisi, ac egestas elit. Suspendisse ultricies scelerisque ipsum, vestibulum cursus sem viverra sit amet. Praesent dapibus eleifend nisl non consequat.
    </p>
  </div>
  <div class="article">
    <h1 class="title">
      updated title [2]
    </h1>
    <p class="author">
      Sangkeun Kim
    </p>
    <p class="content">
      Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed eu ipsum elit. Sed at fermentum purus. Mauris at magna placerat, porttitor augue at, dictum sem. Donec sodales neque nisl. Aenean ipsum ipsum, euismod a turpis id, imperdiet luctus elit. Donec eleifend lacus vel eros sagittis facilisis. Praesent at porttitor erat. Nulla facilisi. Aenean et hendrerit nisi, ac egestas elit. Suspendisse ultricies scelerisque ipsum, vestibulum cursus sem viverra sit amet. Praesent dapibus eleifend nisl non consequat.
    </p>
  </div>
  <div class="article">
    <h1 class="title">
      updated title [3]
    </h1>
    <p class="author">
      Sangkeun Kim
    </p>
  </div>
</div>
```

## updated title [1]

Sangkeun Kim

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed eu ipsum elit. Sed at fermentum purus. Mauris at magna placerat, porttitor augue at, dictum sem. Donec sodales neque nisl. Aenean ipsum ipsum, euismod a turpis id, imperdiet luctus elit. Donec eleifend lacus vel eros sagittis facilisis. Praesent at porttitor erat. Nulla facilisi. Aenean et hendrerit nisi, ac egestas elit. Suspendisse ultricies scelerisque ipsum, vestibulum cursus sem viverra sit amet. Praesent dapibus eleifend nisl non consequat.

## updated title [2]

Sangkeun Kim

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed eu ipsum elit. Sed at fermentum purus. Mauris at magna placerat, porttitor augue at, dictum sem. Donec sodales neque nisl. Aenean ipsum ipsum, euismod a turpis id, imperdiet luctus elit. Donec eleifend lacus vel eros sagittis facilisis. Praesent at porttitor erat. Nulla facilisi. Aenean et hendrerit nisi, ac egestas elit. Suspendisse ultricies scelerisque ipsum, vestibulum cursus sem viverra sit amet. Praesent dapibus eleifend nisl non consequat.

## updated title [3]

Sangkeun Kim

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed eu ipsum elit. Sed at fermentum purus. Mauris at magna placerat, porttitor augue at, dictum sem. Donec sodales neque nisl. Aenean ipsum ipsum, euismod a turpis id, imperdiet luctus elit. Donec eleifend lacus vel eros sagittis facilisis. Praesent at porttitor erat. Nulla facilisi. Aenean et hendrerit nisi, ac egestas elit. Suspendisse ultricies scelerisque ipsum, vestibulum cursus sem viverra sit amet. Praesent dapibus eleifend nisl non consequat.

```
html, body {  
  padding: 0;  
  margin: 0;  
  width: 100%;  
  height: 100%;  
}  
h1, h2, h3, h4, h5, p {  
  margin: 0;  
}  
.navbar-wrapper {  
  height: 50px;  
  background: #24292e;  
  color: hsla(0, 0%, 100%, .7);  
}  
.navbar-container {  
  width: 60%;  
  height: 50px;  
  margin: 0 auto;  
}  
.logo {  
  font-size: 20pt;  
  padding-top: 10px;  
}  
.main-wrapper {  
  width: 100%;  
}  
.main-container {  
  width: 70%;  
  margin: 0 auto;  
}  
.article-container {  
  width: 100%;  
  padding-top: 20px;  
}  
.article {  
  margin-top: 20px;  
  border: 1px solid #d1d5da;  
  padding: 10px;  
  width: 100%;  
}
```

과제 CSS 완성

github 를 참고해.

.article

.article .title

.article .author

.article .content

article 내부의 class에 접근!!

## updated title [1]

Sangkeun Kim

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed eu ipsum elit. Sed at fermentum purus. Mauris at magna placerat, porttitor augue at, dictum sem. Donec sodales neque nisl. Aenean ipsum ipsum, euismod a turpis id, imperdiet luctus elit. Donec eleifend lacinia vel eros sagittis facilisis. Praesent at porttitor erat. Nulla facilisi. Aenean et hendrerit nisi, ac egestas elit. Suspendisse ultricies scelerisque ipsum, vestibulum cursus sem viverra sit amet. Praesent dapibus eleifend nisl non consequat.

## updated title [2]

Sangkeun Kim

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed eu ipsum elit. Sed at fermentum purus. Mauris at magna placerat, porttitor augue at, dictum sem. Donec sodales neque nisl. Aenean ipsum ipsum, euismod a turpis id, imperdiet luctus elit. Donec eleifend lacinia vel eros sagittis facilisis. Praesent at porttitor erat. Nulla facilisi. Aenean et hendrerit nisi, ac egestas elit. Suspendisse ultricies scelerisque ipsum, vestibulum cursus sem viverra sit amet. Praesent dapibus eleifend nisl non consequat.

## updated title [3]

Sangkeun Kim

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed eu ipsum elit. Sed at fermentum purus. Mauris at magna placerat, porttitor augue at, dictum sem. Donec sodales neque nisl. Aenean ipsum ipsum, euismod a turpis id, imperdiet luctus elit. Donec eleifend lacinia vel eros sagittis facilisis. Praesent at porttitor erat. Nulla facilisi. Aenean et hendrerit nisi, ac egestas elit. Suspendisse ultricies scelerisque ipsum, vestibulum cursus sem viverra sit amet. Praesent dapibus eleifend nisl non consequat.

# javascript?

## javascript

1. 동적으로 contents를 다룰 수 있음
2. 이벤트 (마우스 click, scroll, hover 등)를 다룰 수 있음
3. asynchronous network 통신을 할 수 있음
  - API 서버로부터 데이터를 가져올 수 있음

# javascript?

## javascript

LOGO

### updated title [1]

Sangkeun Kim

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed eu ipsum elit. Sed at fermentum purus. Mauris at magna placerat, porttitor augue at, dictum sem. Donec sodales neque nisl. Aenean ipsum ipsum, euismod a turpis id, imperdiet luctus elit. Donec eleifend lacus vel eros sagittis facilisis. Praesent at porttitor erat. Nulla facilisi. Aenean et hendrerit nisi, ac egestas elit. Suspendisse ultricies scelerisque ipsum, vestibulum cursus sem viverra sit amet. Praesent dapibus eleifend nisl non consequat.

### updated title [2]

Sangkeun Kim

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed eu ipsum elit. Sed at fermentum purus. Mauris at magna placerat, porttitor augue at, dictum sem. Donec sodales neque nisl. Aenean ipsum ipsum, euismod a turpis id, imperdiet luctus elit. Donec eleifend lacus vel eros sagittis facilisis. Praesent at porttitor erat. Nulla facilisi. Aenean et hendrerit nisi, ac egestas elit. Suspendisse ultricies scelerisque ipsum, vestibulum cursus sem viverra sit amet. Praesent dapibus eleifend nisl non consequat.

### updated title [3]

Sangkeun Kim

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed eu ipsum elit. Sed at fermentum purus. Mauris at magna placerat, porttitor augue at, dictum sem. Donec sodales neque nisl. Aenean ipsum ipsum, euismod a turpis id, imperdiet luctus elit. Donec eleifend lacus vel eros sagittis facilisis. Praesent at porttitor erat. Nulla facilisi. Aenean et hendrerit nisi, ac egestas elit. Suspendisse ultricies scelerisque ipsum, vestibulum cursus sem viverra sit amet. Praesent dapibus eleifend nisl non consequat.

형태 : HTML

디자인 : CSS

① API 서버로부터 데이터 받아서  
보여주고 싶을 때는? → 통신, 동작 contents

② 제목을 click하면 Article의  
Summary를 간단히 보여줘! → event

③ 카드는 단축키로 사용자가 편했으면  
좋겠어! → event

# javascript?

## javascript

```
</body>
<script>
  const logoElement = document.getElementById('logo');
  const f = (e) => {
    console.log(e);
    alert('You clicked the logo');
    if (confirm('If you click ok button, then this page will be refreshed')) {
      window.location.reload();
    } else {
      alert('You clicked no! button :)');
    }
  }
  logoElement.addEventListener('click', f);

  const elements = document.getElementsByClassName('title');
  for (let i = 0; i < elements.length; i++) {
    elements[i].addEventListener('click', (e) => {
      alert(e.target.innerText);
    });
  }
</script>
</html>
```

logo click

title click

The image displays two screenshots of a web browser. The top screenshot shows a single logo element with a yellow arrow pointing to it from the handwritten text 'logo click'. The bottom screenshot shows multiple title elements with a yellow arrow pointing to it from the handwritten text 'title click'. Both screenshots show an alert box with the message 'You clicked the logo' or 'updated title [1]'. The browser interface includes a header with 'LOGO', a main content area with placeholder text, and a footer with a close button.

LOGO

updated title [1]  
Sangkeun Kim  
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed eu ipsum elit. Sed at fermentum purus. Mauris at magna placerat, porttitor augue at, dictum sem. Donec sodales neque nisl. Aenean ipsum ipsum, euismod a turpis id, imperdier luctus elit. Donec eleifend lacus vel eros sagittis facilisis. Praesent at porttitor erat. Nulla facilisi. Aenean et hendrerit nisi, ac egestas elit. Suspendisse ultricies scelerisque ipsum, vestibulum cursus sem viverra sit amet. Praesent dapibus eleifend nisl non consequat.

updated title  
Sangkeun Kim  
You clicked the logo  
닫기

updated title [3]  
Sangkeun Kim  
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed eu ipsum elit. Sed at fermentum purus. Mauris at magna placerat, porttitor augue at, dictum sem. Donec sodales neque nisl. Aenean ipsum ipsum, euismod a turpis id, imperdier luctus elit. Donec eleifend lacus vel eros sagittis facilisis. Praesent at porttitor erat. Nulla facilisi. Aenean et hendrerit nisi, ac egestas elit. Suspendisse ultricies scelerisque ipsum, vestibulum cursus sem viverra sit amet. Praesent dapibus eleifend nisl non consequat.

LOGO

updated title [1]  
Sangkeun Kim  
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed eu ipsum elit. Sed at fermentum purus. Mauris at magna placerat, porttitor augue at, dictum sem. Donec sodales neque nisl. Aenean ipsum ipsum, euismod a turpis id, imperdier luctus elit. Donec eleifend lacus vel eros sagittis facilisis. Praesent at porttitor erat. Nulla facilisi. Aenean et hendrerit nisi, ac egestas elit. Suspendisse ultricies scelerisque ipsum, vestibulum cursus sem viverra sit amet. Praesent dapibus eleifend nisl non consequat.

updated title [2]  
Sangkeun Kim  
updated title [2]  
닫기

updated title [3]  
Sangkeun Kim  
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed eu ipsum elit. Sed at fermentum purus. Mauris at magna placerat, porttitor augue at, dictum sem. Donec sodales neque nisl. Aenean ipsum ipsum, euismod a turpis id, imperdier luctus elit. Donec eleifend lacus vel eros sagittis facilisis. Praesent at porttitor erat. Nulla facilisi. Aenean et hendrerit nisi, ac egestas elit. Suspendisse ultricies scelerisque ipsum, vestibulum cursus sem viverra sit amet. Praesent dapibus eleifend nisl non consequat.

# javascript

## Coding Convention

1. Camel case type language
2. 2-space indentation
3. Do not use quotes around key of object
4. global variable names written in UPPERCASE

# javascript

## Variable, Types, list, dictionary

1. var: 가장 기본 타입, 변수 재선언 가능, function-scope
2. let: 변수 재선언 불가능, mutable
3. const: immutable

\* var를 멀리하고 let과 const를 쓰면 됩니다

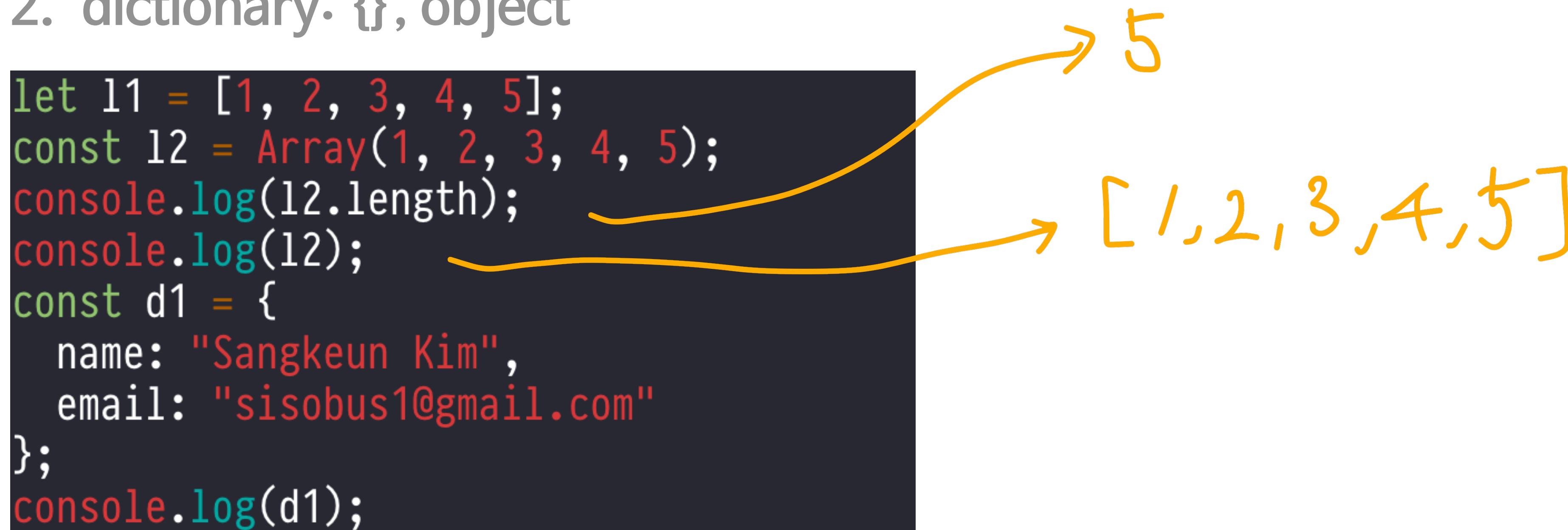
```
var a = 5;
var a = 'test';
let b = 6;
// let b = 7; // this statement is error
const c = 5;
// c = 7; // this statement is error (c is immutable)
```

# javascript

## Variable, Types, list, dictionary

1. list: [], Array
2. dictionary: {}, object

```
let l1 = [1, 2, 3, 4, 5];
const l2 = Array(1, 2, 3, 4, 5);
console.log(l2.length);
console.log(l2);
const d1 = {
  name: "Sangkeun Kim",
  email: "sisobus1@gmail.com"
};
console.log(d1);
```



# javascript

if

```
const pi = 3.1415927584;
if (pi > 3 && pi < 4) {
  console.log(`pi is in range(${3}, ${4})`);
} else if (pi <= 3 || pi >= 4) {
  console.log('pi is not in range(3, 4)');
} else {
  console.log('It is impossible');
}
```

# javascript

for

```
let l = [1, 2, 3, 4, 5];
for (let i = 0; i < l.length; i++) {
  console.log(l[i]);
}
for (let i = 0; i < 10; i++) {
  l.push(i);
}
```

# javascript

## function

1. 두 가지 문법이 있음

1. function: 일반적인 함수

2. arrow function: 익명 함수

→  $(\text{param}_1, \dots, \text{param}_N) \Rightarrow \{\text{statements}\}$

①  $(\sim) \Rightarrow \text{expression}$

$\{\text{return expression}\}$

②  $\text{singleParam} \Rightarrow \text{expression}$

$(\text{singleParam}) \Rightarrow \{\text{return expression}\}$

③  $() \Rightarrow \{\text{statements}\}$

```
function f(a, b, c) {  
    a = a + 1;  
    b.push(5);  
    c['test'] = "T_T";  
}  
let _a = 5;  
let b = [4, 3];  
let c = {};  
f(_a, b, c);  
console.log(_a, b, c); // 5 [4, 3, 5] {test: "T_T"}
```

```
const ff = (a, b, c) => {  
    a = a + 1;  
    b.push(5);  
    c['test'] = "T_T";  
}  
let _A = 5;  
let B = [4, 3];  
let C = {};  
ff(_A, B, C);  
console.log(_A, B, C); // 5 [4, 3, 5] {test: "T_T"}
```

# javascript

## class

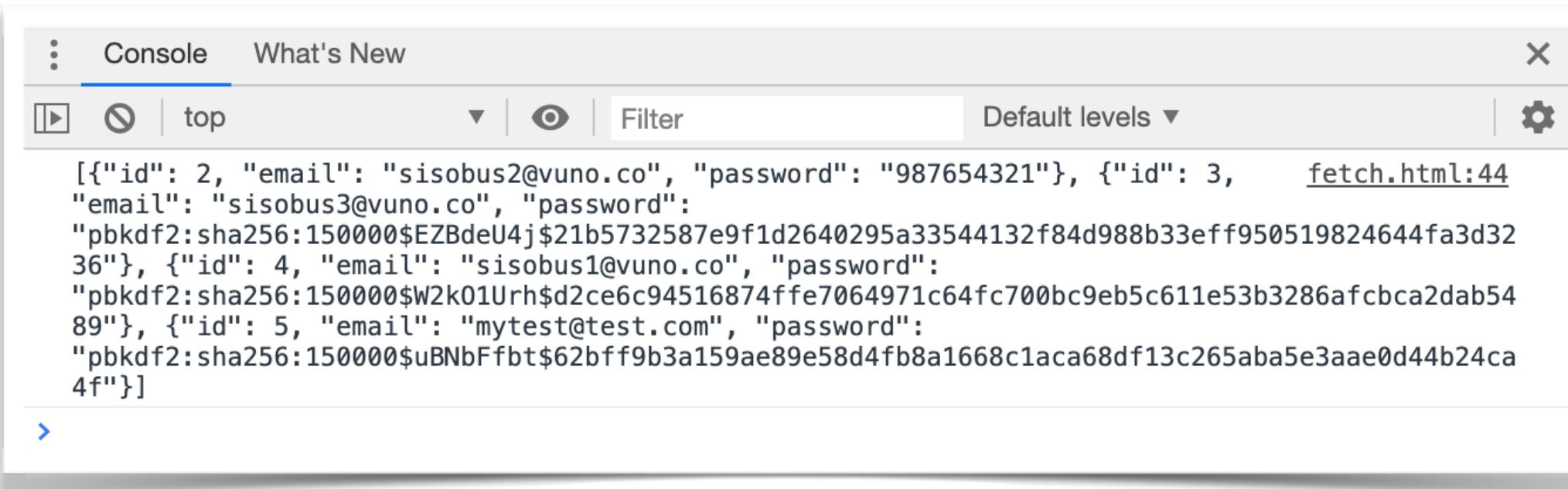
1. method에는 function을 명시하지 않음
2. constructor는 instance화 될 때 실행됨
3. 상속은 extends를 이용함

```
class Wine {  
    constructor() {  
        this.isAlcohol = true;  
        console.log("This is constructor");  
    }  
    color() {  
        return 'Red or white or Sparkling';  
    }  
}  
class RedWine extends Wine {  
    color() {  
        return 'Red';  
    }  
}  
wine = new Wine();  
console.log(wine.isAlcohol, wine.color());  
redWine = new RedWine();  
console.log(redWine.isAlcohol, redWine.color());  
/*  
 * This is constructor  
 * true "Red or white or Sparkling"  
 * This is constructor  
 * true "Red"  
 */
```

# javascript!

## fetch

1. 서버와 통신할 때 쓰임
2. 비동기 통신
3. Promise를 반환함

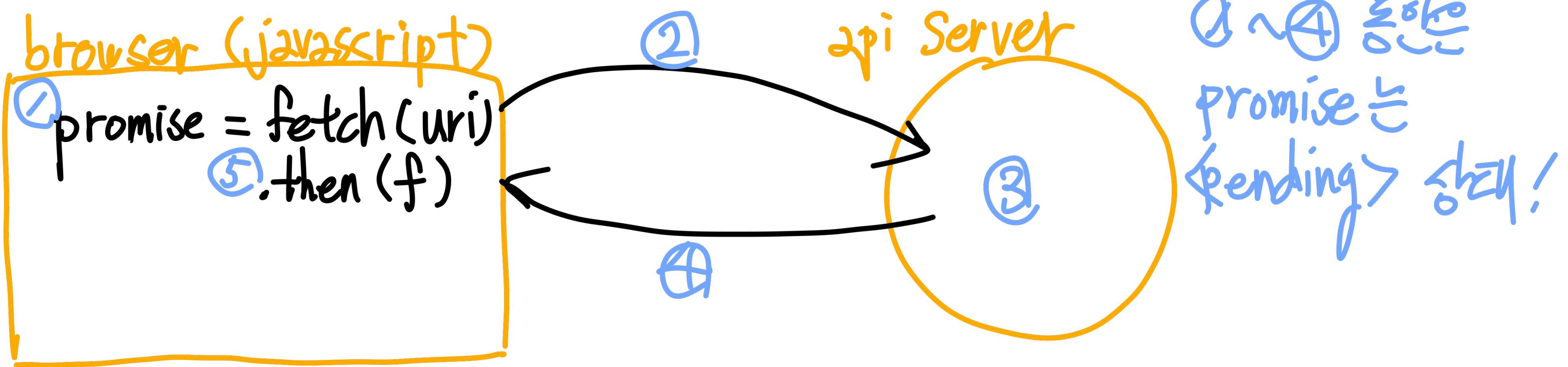


```
let requestOptions = {
  method: 'GET',
  headers: { 'Content-Type': 'application/json' }
}
const handleResponse = response => {
  return response.text().then(text => {
    const data = text && JSON.parse(text)
    if (!response.ok) {
      if (response.status === 401) {
        // auto logout if 401 response returned from api
        return Promise.reject(response)
      }
      const error = (data && data.message) || response.statusText
      return Promise.reject(error)
    }
    return data
  })
}
fetch('http://0.0.0.0:5000/api/users', requestOptions)
  .then(handleResponse)
  .then(response => {
    console.log(response);
  })
  .catch(error => {
    console.log(error);
  });
});
```

# javascript!

## promise

1. 서버에 요청하면 기다리게됨 (정말? 얼마나 기다리지?)
2. Javascript는 Single thread 기반 언어이기 때문에 화면이 멈춤 => 이럼 안됨
3. Promise는 비동기 호출
4. 언제 끝날지는 모르겠지만 끝나면 등록한 함수를 실행해줘! => Callback function
5. 서버의 응답이 오면 Callback function이 실행 됨



# javascript!

## CORS (Cross-Origin Resource Sharing)

1. fetch에서 아마 에러가 날 것임
2. javascript에서 외부 IP 주소로 fetch를 했기 때문
3. API 서버에서 cors 처리를 해줘야함
  1. 라이브러리 설치 \$ pip install flask\_cors
  2. flask 서버에 적용

1. from flask\_cors import CORS

2. app = Flask(\_\_name\_\_)

3. cors = CORS(app)

```
from flask import Flask, request
from flask_restful import Api, Resource
from flask_cors import CORS
app = Flask(__name__)
cors = CORS(app)
api = Api(app)
db.init_app(app)
```

# javascript!

## POST method

1. body에 데이터를 json형식으로 Serialize해서 보냄

A screenshot of a browser's developer tools Console tab. The tab title is 'Console' and it shows the following output:

```
{id: 2, "email": "sisobus2@vuno.co", "password": "987654321"} {"id": 3, "email": "sisobus3@vuno.co", "password": "pbkdf2:sha256:150000$EZBdeU4j$21b5732587e9f1d2640295a33544132f84d988b33eff950519824644fa3d3236"} {"id": 4, "email": "sisobus1@vuno.co", "password": "pbkdf2:sha256:150000$W2k01Urh$d2ce6c94516874ffe7064971c64fc700bc9eb5c611e53b3286afcbca2dab5489"}  
create email: mytest@test.com, pw: mytest successfully
```

```
requestOptions = {  
  method: 'POST',  
  headers: { 'Content-Type': 'application/json' },  
  body: JSON.stringify({  
    email: "mytest@test.com",  
    password: "mytest"  
  })  
}  
fetch('http://0.0.0.0:5000/api/users', requestOptions)  
  .then(handleResponse)  
  .then(response => {  
    console.log(response);  
  })  
  .catch(error => {  
    console.log(error);  
});
```

# javascript!

## POST method

1. body에 데이터를 json형식으로 Serialize해서 보냄

A screenshot of a browser's developer tools Console tab. The tab title is 'Console' and it shows the following output:

```
{id: 2, "email": "sisobus2@vuno.co", "password": "987654321"} {"id": 3, "email": "sisobus3@vuno.co", "password": "pbkdf2:sha256:150000$EZBdeU4j$21b5732587e9f1d2640295a33544132f84d988b33eff950519824644fa3d3236"} {"id": 4, "email": "sisobus1@vuno.co", "password": "pbkdf2:sha256:150000$W2k01Urh$d2ce6c94516874ffe7064971c64fc700bc9eb5c611e53b3286afcbca2dab5489"}  
create email: mytest@test.com, pw: mytest successfully
```

```
requestOptions = {  
  method: 'POST',  
  headers: { 'Content-Type': 'application/json' },  
  body: JSON.stringify({  
    email: "mytest@test.com",  
    password: "mytest"  
  })  
}  
fetch('http://0.0.0.0:5000/api/users', requestOptions)  
  .then(handleResponse)  
  .then(response => {  
    console.log(response);  
  })  
  .catch(error => {  
    console.log(error);  
});
```

# javascript!

## Article fetch 후 보여주기

1. 데이터를 fetch 해온 뒤
2. html로 만들고
3. articleContainer 안에  
insertAdjacentHTML 메소드로  
삽입

```
fetch('http://0.0.0:5000/api/articles', requestOptions)
  .then(handleResponse)
  .then(response => {
    response = JSON.parse(response);
    let html = "";
    for (let i = 0; i < response.length; i++) {
      html += `<div class="article" id="article-${response[i].id}">`;
      html += `<h1 class="title">${response[i].title}</h1>`;
      html += `<p class="author">${response[i].author}</p>`;
      html += `<p class="content">${response[i].content}</p>`;
      html += `</div>`;
    }
    const articleContainer = document.getElementById('article-container');
    articleContainer.insertAdjacentHTML('beforeEnd', html);
  })
  .catch(error => {
    console.log(error);
});
```

## updated title

sisobus3@vuno.co

updated content

## test title3

sisobus3@vuno.co

test content3

## test title4

sisobus2@vuno.co

test content4

## test title1

sisobus2@vuno.co

test content1

## test title2

sisobus3@vuno.co

test content2

## test title3

sisobus3@vuno.co

# Summary

## 정리!

1. HTML, CSS: 페이지의 윤곽, 디자인
2. javascript: 데이터 통신, 이벤트 처리 등
3. 6\_relationship까지 만들었던 api server를 켜놔야 함
4. server에서 CORS 처리를 해줘야함
5. (serializer 수정해야함, 내 코드는 되어있음)
6. fetch는 비동기 통신, promise를 반환함
7. document.getElementById로 부모 element를 선택 후 그 안에
8. insertAdjacentHTML('beforeEnd', HTML); 로 동적으로 삽입 할 수 있음

# 실습

## branch: 8-practice-〈UsernameEn〉

1. API Server 켜놓기
2. tests.sh로 데이터 삽입하기
3. index.html, fetch.html 실행해보기
4. article title 클릭해서 글 삭제 구현하기 (이거 어려움)
5. 질문은 issues에다가
6. 데드라인은 수요일
7. 답지는 화요일 점심에 올라옴

# Q & A