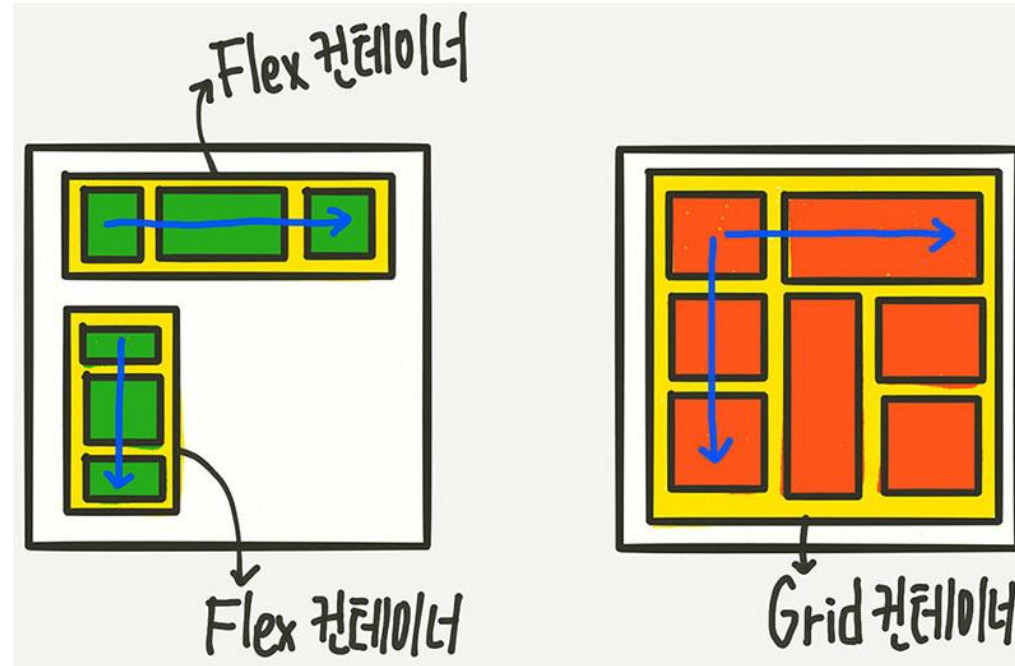


프론트 세미나 Week4  
Responsive Web Design &  
JS 문법

# Review



Last lecture: CSS Position & Layout

- 화면 크기가 줄어들어도 내용이 변하지 않는 문제점! → 모바일 화면에서 불편할 수 있음!

# Responsive Web Design









# Responsive Web Design



- 화면 크기에 따라 element들의 위치가 변경됨!
- chrome 개발자 도구를 통해 모바일, 태블릿 또는 기종별 화면 설정 가능

# Responsive Web Design



Result Size: 633 x 461

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>

<h2>Responsive Image</h2>
<p>"max-width:100%" prevents the image from getting bigger than its
original size. However, if you make the browser window smaller, the
image will still scale down.</p>
<p>Resize the browser window to see the effect.</p>




</body>
</html>
```

## Responsive Image

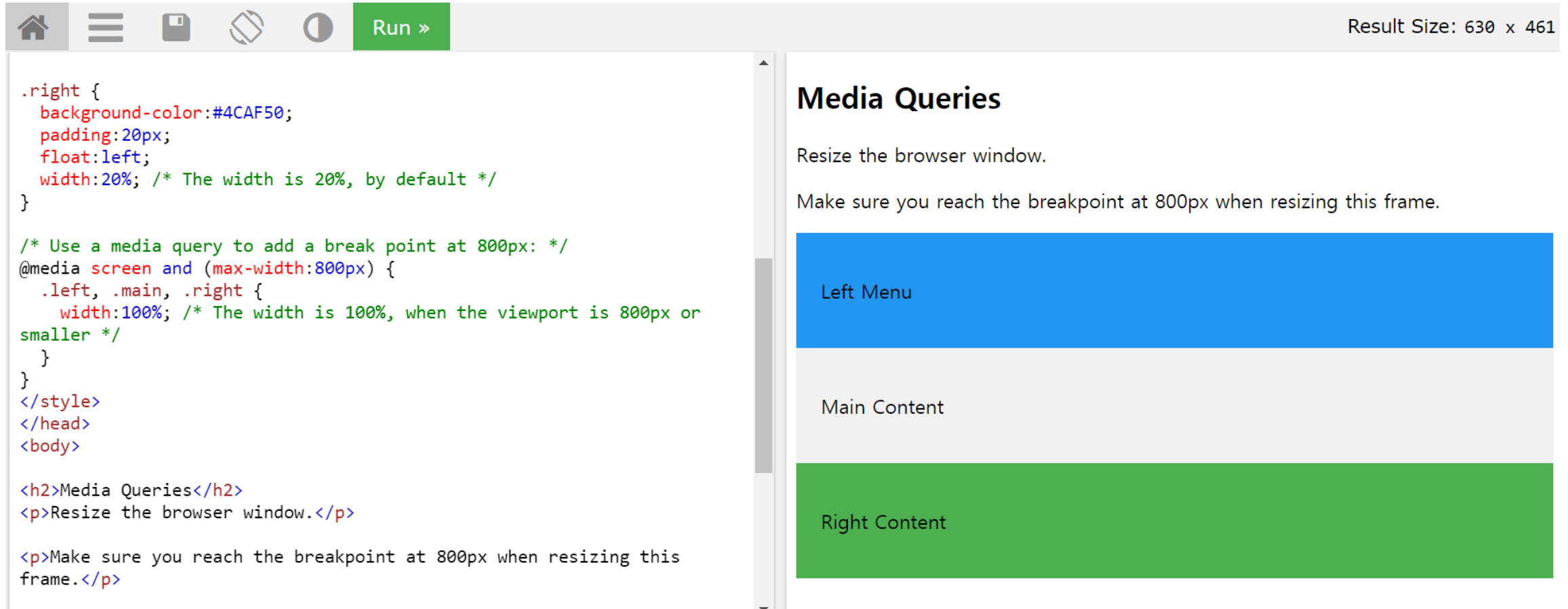
"max-width:100%" prevents the image from getting bigger than its original size. However, if you make the browser window smaller, the image will still scale down.

Resize the browser window to see the effect.



- 이전: 창 크기에 따라 이미지 크기 변동
- max-width: 100%: 창 크기가 커져도 최대 크기 = 원본 size

# Responsive Web Design



The screenshot shows a web browser interface with a toolbar at the top containing icons for home, menu, save, print, and a 'Run' button. The result size is indicated as 630 x 461. The browser displays a web page with a blue header, a light gray main content area, and a green right sidebar. The CSS code in the left pane defines the layout using media queries to change the width of the sidebar from 20% to 100% at an 800px breakpoint.

```
.right {  
  background-color:#4CAF50;  
  padding:20px;  
  float:left;  
  width:20%; /* The width is 20%, by default */  
}  
  
/* Use a media query to add a break point at 800px: */  
@media screen and (max-width:800px) {  
  .left, .main, .right {  
    width:100%; /* The width is 100%, when the viewport is 800px or  
smaller */  
  }  
}  
</style>  
</head>  
<body>  
  
<h2>Media Queries</h2>  
<p>Resize the browser window.</p>  
  
<p>Make sure you reach the breakpoint at 800px when resizing this  
frame.</p>
```

## Media Queries

Resize the browser window.

Make sure you reach the breakpoint at 800px when resizing this frame.

Left Menu

Main Content

Right Content

- media query 사용(@media (screen and) ) -> 윈도우 크기에 따라 CSS property 다르게 부여
- max-width or min-width: 최대/최소 창 크기 설정
- width<=max-width or width>=min-width 일 때만 @media 안의 내용 실행

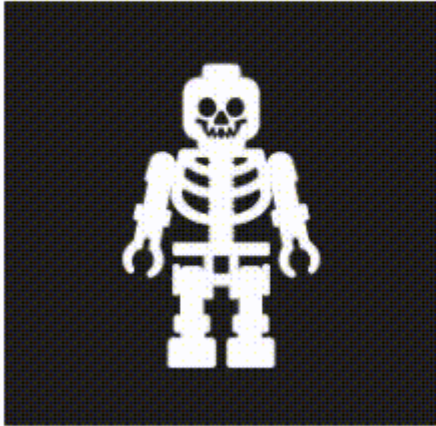
# Responsive Web Design



- media query를 이용하면 Layout을 좀 더 업그레이드 가능

# HTML, CSS, JS

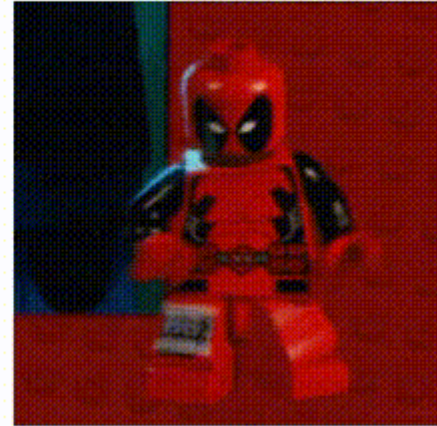
**HTML**  
structure



**CSS**  
presentation/appearance



**JavaScript**  
dynamism/action





# HTML, CSS, JS

네이버를 시작페이지로 > | [주니어네이버](#) [해피빈](#)



NAVER

더보기 ▾

☀ 11.0° 맑음 13.0° / 19.0° 재송동

연합뉴스 > 발동 걸린 '부동산 적폐청산'...숨죽인 기득권층

네이버뉴스 · 연예 스포츠 경제

뉴스스탠드 > 구독한 언론사 · 전체언론사


네이버를 더 안전하고 편리하게 이용하세요

NAVER 로그인

🔒 아이디 · 비밀번호찾기

👤 회원가입

LIVE 저녁방송 메인뉴스 보기



트렌드쇼핑 >

상품 쇼핑물 MEN

G마켓 · 옥션 · 11번가 · 위메프 · 이마트몰 · 티몬  
신세계몰 · 올리브영 · GS샵 · 롯데이몰 · 롯데ON · Cmall

🌙 다크 모드로 보기



# Before We Start...

- JS의 모든 문법을 이 시간에 다룰 수는 없으므로 핵심 부분만 간략하게 언급하고 넘어갈 것입니다! 세미나에서 언급되지 않은 부분은 코드 작성 시마다 구글링해나가며 찾아보길 권해드립니다!
- 본 시간에는 C, C++, Python 등 익숙한 언어에 비유하여 문법을 설명할 것입니다! 프로그래밍 언어를 한 번도 접해본 적이 없거나 본 세미나에서 비유하는 개념을 이해하기 어려울 경우 따로 관련 개념을 찾아보고 학습하시길 권해드립니다!
- 백엔드 세미나를 수강하셨거나 다른 경로로 JS 문법을 이미 이해하고 계신 경우 복습 차원으로 가볍게 들어주시면 감사하겠습니다!

# Data type

## 데이터 타입

최신 ECMAScript 표준은 다음과 같은 7개의 자료형을 정의한다.

- [기본 자료형 \(Primitive\)](#) 인 여섯가지 데이터 타입
  - [Boolean](#)
  - [Null](#)
  - [Undefined](#)
  - [Number \(en-US\)](#)
  - [String](#)
  - [Symbol](#) (ECMAScript 6 에 추가됨)
- 별도로 [Object](#) 도 있음

- Boolean: True/ False
- Null: invalid, 비어있는 값
- Undefined: 값을 할당하지 않은 상태
- Number: int, float을 포괄하는 data type, C에서의 float처럼 +inf, -inf, NaN도 포함
- String: 문자열, 텍스트 데이터
- 변수의 data type을 알기 위해서는 typeof 연산자 이용

# Data type

```
var x = 5;  
var y = 6;  
var z = x + y;
```

- Javascript에서 변수를 선언하는 3가지: var, let, const
- C에서와 달리 data type을 명시할 필요 x
- var: 기본적인 변수 선언
- let: 변수 재선언 불가능
- const: 값의 재할당 불가능(상수 취급)
- JS에선 세미콜론을 적어도 되고, 적지 않아도 됨(컴파일러가 자동적으로 줄바꿈 시 세미콜론 삽입)

ex)

var x=5;	let x=5;	const x=5;
var x=3; (o)	let x=3; (x)	const x=3; (x)
x=3; (o)	x=3; (o)	x=3; (x)

# Array

```
var cars = ["Saab", "Volvo", "BMW"];
```

---

```
var name = cars[0];
```

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];  
fruits.push("Lemon");    // adds a new element (Lemon) to fruits
```

- python의 list와 유사
- 배열 길이 반환 -> .length ex) cars.length=3
- 새 element 추가: push

# Comparison/Conditions

Operator	Description	Comparing	Returns
==	equal to	x == 8	false
		x == 5	true
		x == "5"	true
===	equal value and equal type	x === 5	true
		x === "5"	false

- == → data type이 달라도 값이 같으면 equal
- === → data type까지 같아야 equal
- 조건문: C와 동일하게 if, else if, else로 처리

# Loop

```
var i;  
for (i = 0; i < cars.length; i++) {  
    text += cars[i] + "<br>";  
}
```

```
while (i < 10) {  
    text += "The number is " + i;  
    i++;  
}
```

- 반복문: C, Python 등과 유사
- for문 또는 while문 등을 사용
- break, continue 등과 같은 것도 사용 가능

# Function


```
function myFunction(p1, p2) {  
    return p1 * p2;    // The function returns the product of p1 and p2  
}
```

```
1  var numbering = function () {  
2      i = 0;  
3      while (i < 10) {  
4          document.write(i);  
5          i += 1;  
6      }  
7  }  
8  numbering();
```

- JS에서의 함수: function 키워드 사용
- 함수를 변수로 취급 가능



# Object

Object	Properties	Methods
	<code>car.name = Fiat</code> <code>car.model = 500</code> <code>car.weight = 850kg</code> <code>car.color = white</code>	<code>car.start()</code> <code>car.drive()</code> <code>car.brake()</code> <code>car.stop()</code>

- 객체: property(변수)와 method(함수)를 가짐

# Object

```
var person = {  
  firstName: "John",  
  lastName : "Doe",  
  id       : 5566,  
  fullName : function() {  
    return this.firstName + " " + this.lastName;  
  }  
};
```

- Javascript의 객체는 Python의 dictionary와 유사
- key: value의 쌍으로 이루어짐

# Object

```
objectName.propertyName
```

```
objectName.methodName()
```

or

```
objectName["propertyName"]
```

- property 접근 ex) car.wheel 또는 car["wheel"]
- method 접근 ex) car.drive()
- this: 객체 자기 자신을 가리키는 지시어

# Reference

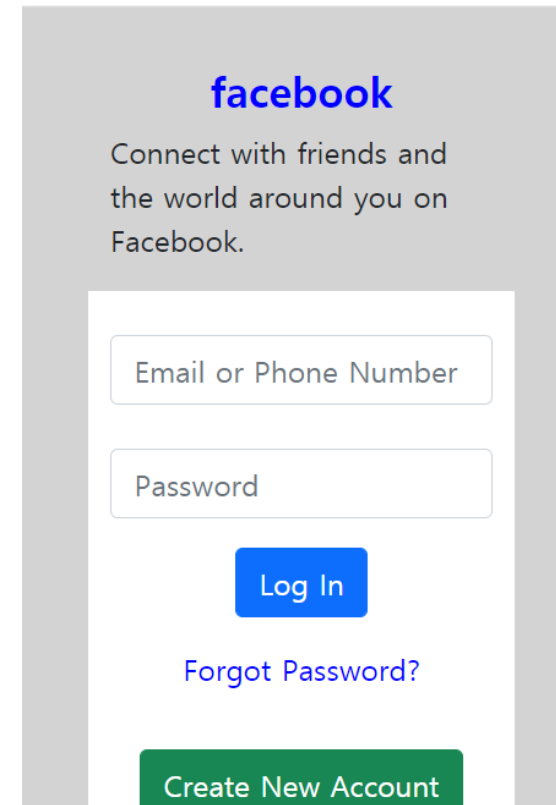
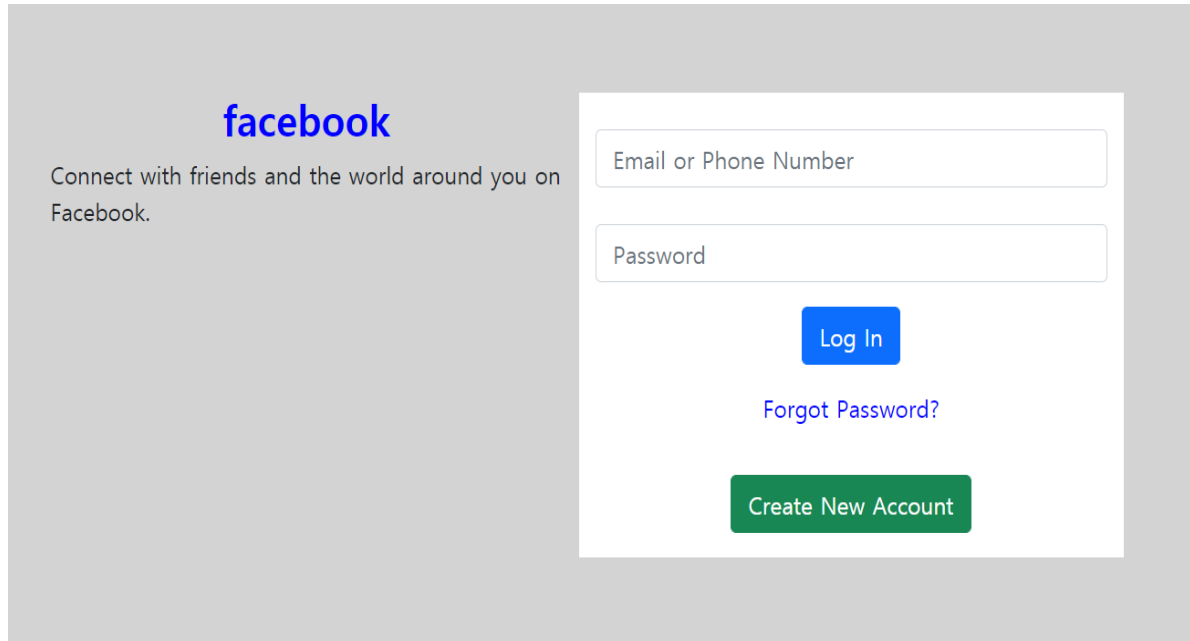
## Responsive Web Design

- [https://www.w3schools.com/html/html\\_responsive.asp](https://www.w3schools.com/html/html_responsive.asp)

## JS

- [https://developer.mozilla.org/ko/docs/Web/JavaScript/Data\\_structures](https://developer.mozilla.org/ko/docs/Web/JavaScript/Data_structures)
- <https://www.w3schools.com/js/default.asp>
- <https://opentutorials.org/course/743>

# 과제



- Responsive Web Design 적용한 페이지 만들기!
- 페이스북 or 인스타그램 or PLMS 등의 메인 화면 클론 코딩!
- Responsive Web Design 요소는 필수, 다른 요소들은 가능한 데까지 최대한 비슷하게 만들어 보기
- 화면을 녹화해서 보내주세요!
- 기한: 다음 수업 전까지!