

JS



{제어를 통해 웹페이지를 풍부하게 해보자!}



1. Javascript를 설명할 수 있다.

2. 기본 문법(변수 ~ 반복문)을 활용할 수 있다.



Javascript

웹 페이지를 동적으로, 프로그래밍적으로 제어하기 위해서
고안된 객체 기반의 스크립트 언어



Javascript 역사

팀 버너스 리
'Web' 발명
HTML 등장

브렌던 아이크



Flash
ActiveX 유행



+
Ajax 방식 등장



1990

1995

2000

2000년대
중·후반

HTML



LiveScript

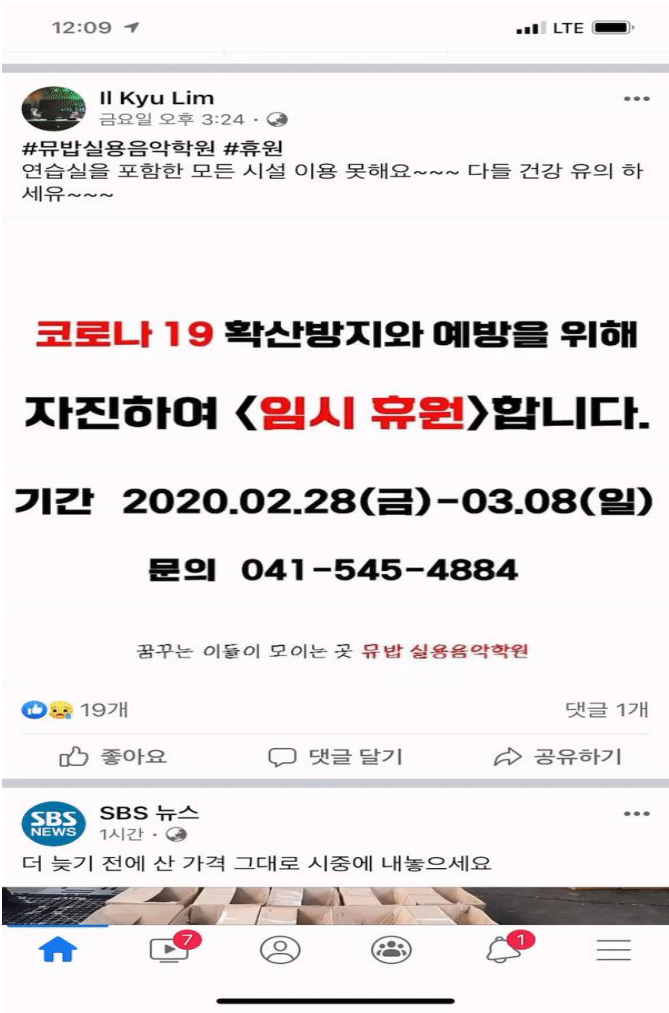


JS





Javascript 어디서 사용될까?



아이디

@naver.com

필수 정보입니다.

비밀번호

필수 정보입니다.

비밀번호 재확인

필수 정보입니다.

이름

필수 정보입니다.

남자

여자

필수 정보입니다.

생일

년(4자)

월

일

태어난 년도 4자리를 정확하게 입력하세요.

비상연락용 이메일

+82

휴대전화번호

인증

필수 정보입니다.

인증번호

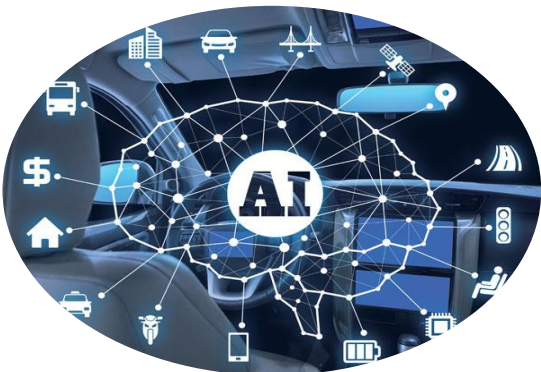
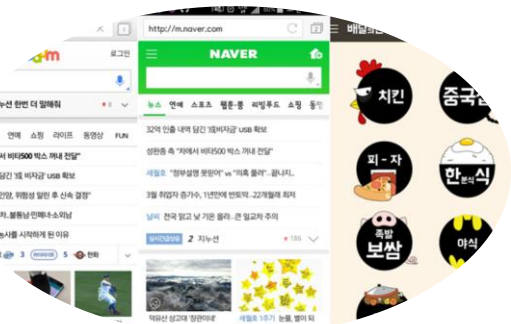
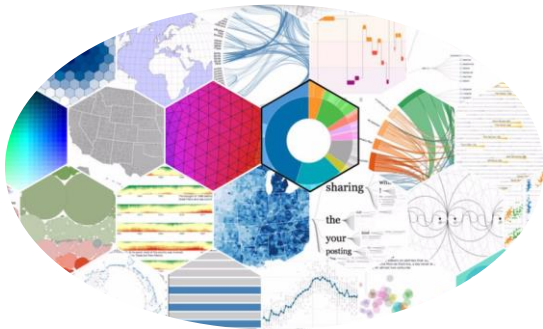
인증이 필요합니다.

확인

입력하신 정보를 다시 확인해주세요.

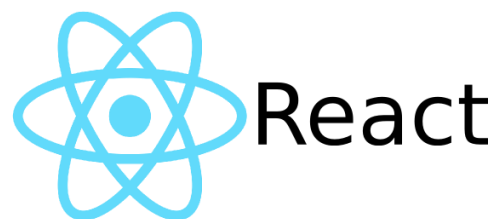


Javascript 활용 분야



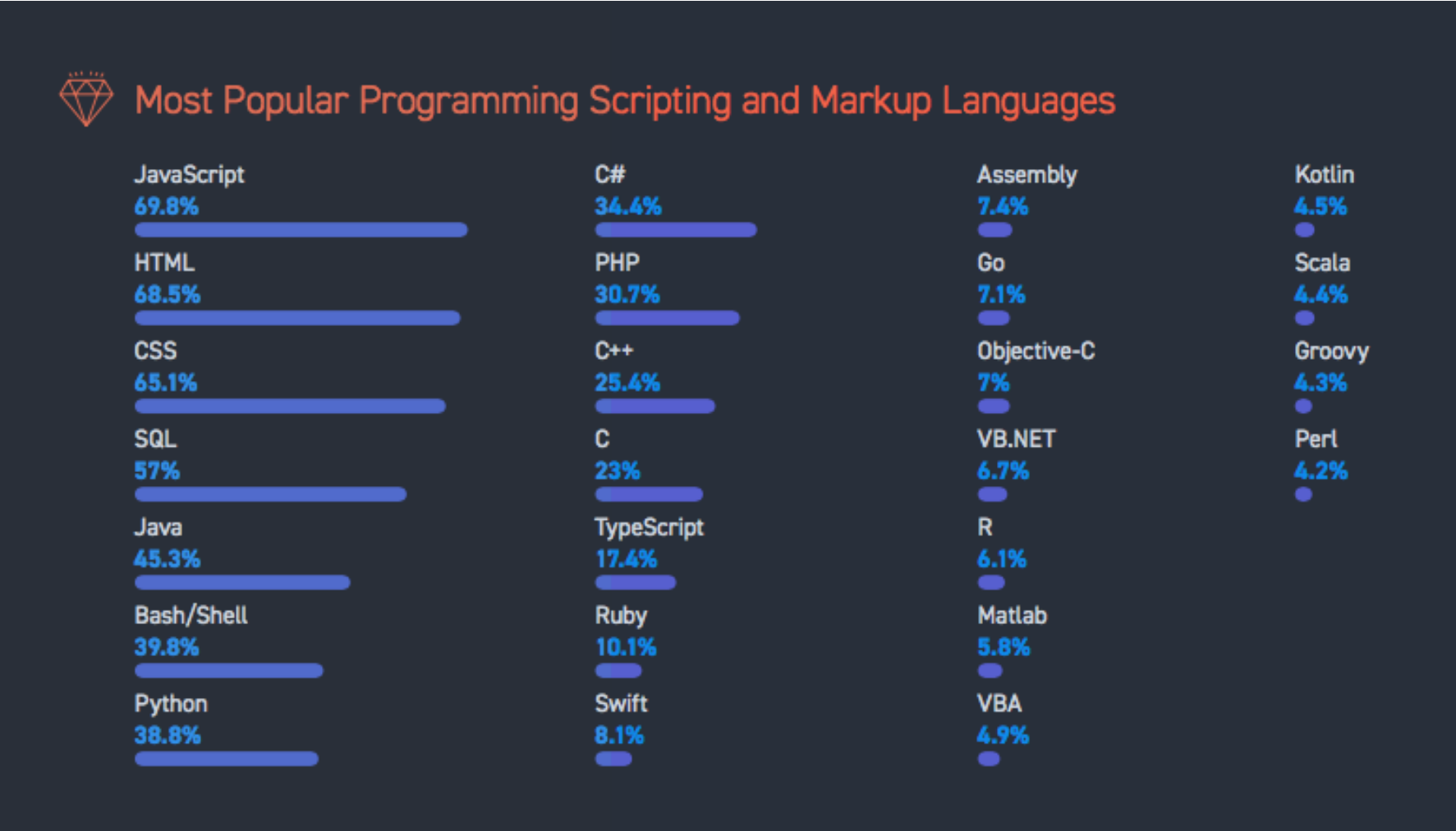


Javascript 관련 기술





1. 모든 웹 브라우저에서 동작한다.
2. 웹 브라우저에서 실행 결과를 즉시 확인
3. 다양한 용도의 프로그램 개발
4. 다양한 자바스크립트 공개 API
5. 다양한 라이브러리와 프레임워크



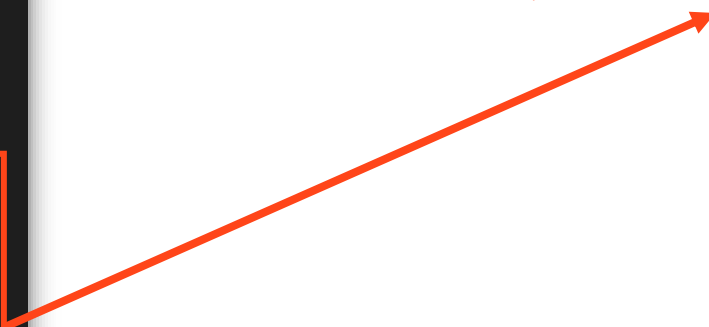


Javascript 사용방법

```
<!DOCTYPE html>
<html>
<head>
  <title>Document</title>
</head>
<body>
  <h1>Javascript</h1>
  <p>Javascript is ....</p>

  <script>
    //Javascript 코드 작성
  </script>
</body>
</html>
```

* </body> 태그 위쪽에 작성





Javascript 입력과 출력

document.write();

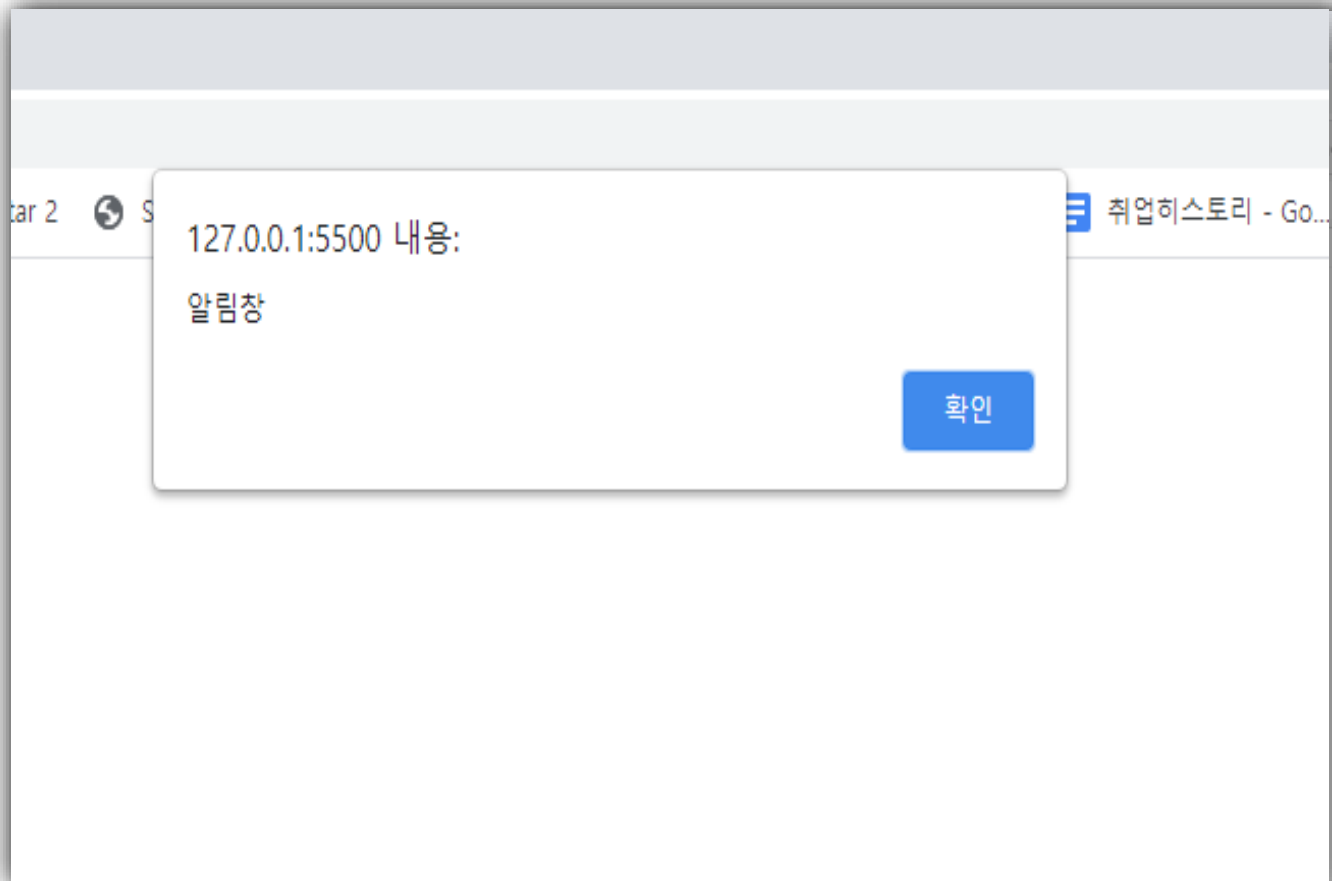
- HTML 문서 내에 출력

console.log();

- console 창에 출력

alert();

- 알림 팝업창으로 출력

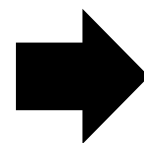





Javascript 출력문

```
> console.log('내용');  
console.info('정보');  
console.warn('경고');  
console.error('오류');
```

console창에 내용입력



내용
정보
 경고
 오류

출력결과



prompt("출력내용", "입력내용")

- 입력창을 통한 입력문

* 리턴타입 : String

localhost:8080 내용:

출력내용

입력내용

확인 취소

confirm("출력문작성")

- 확인 및 취소를 통한 입력문

* 리턴타입: boolean

localhost:8080 내용:

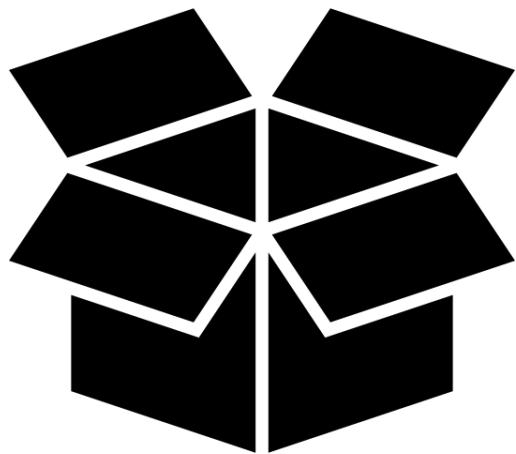
진행하겠습니까?

확인 취소



* 변수 **var** 키워드

- 사전적 의미로는 “변화를 줄 수 있는” 또는 “변할 수 있는 수”
- 프로그래밍에서는 데이터를 담을 수 있는 공간





* 저장하는 데이터에 따라서 자료형 결정

var num = 3;

↓
변수 선언
키워드

↓
변수 명

↓
대입

↓
값





Java vs Javascript 차이

```
int age = 20;  
char alpha = 'a';  
double avg = 78.4;
```

강력한 자료형 체크

```
var age = 20;  
var alpha = 'a';  
var avg = 35.5;
```

느슨한 자료형 체크



ECMAScript6 표준안에서 새로운 키워드 등장

(European Computer Manufacturers Association)

let

변수

const

상수



var vs let vs const

var

```
var aValue = 1;  
var aValue = 2; // OK
```

```
var aValue = 1;  
aValue = 2; // OK
```

1. 변수 재선언 **가능**
2. 변수 재할당 **가능**

let

```
let aValue = 1;  
let aValue = 2; // Error
```

```
let aValue = 1;  
aValue = 2; // OK
```

1. 변수 재선언 **불가능**
2. 변수 재할당 **가능**

const

```
const aValue = 1;  
const aValue = 2; // Error
```

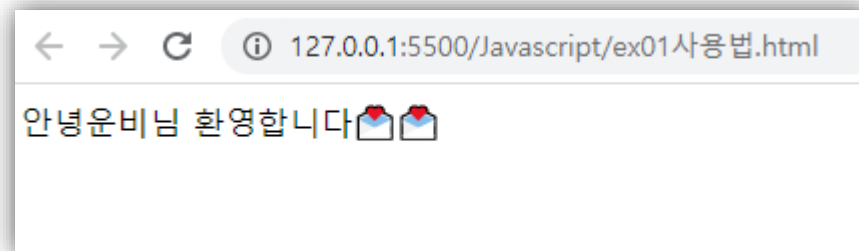
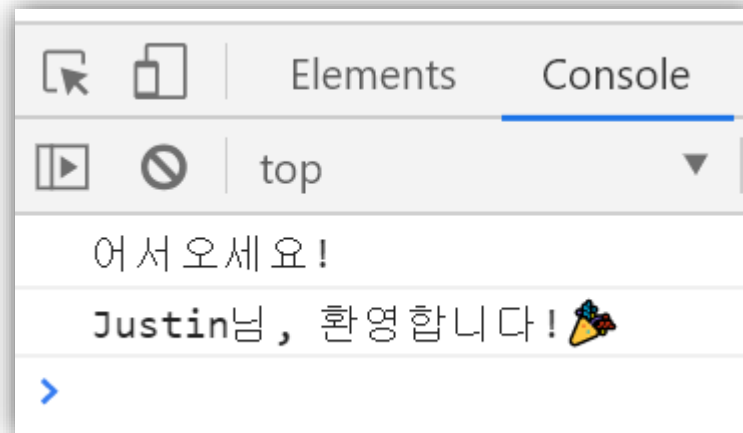
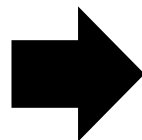
```
const aValue = 1;  
aValue = 2; // error
```

1. 변수 재선언 **불가능**
2. 변수 재할당 **불가능**



입출력문과 변수를 활용한 실습

127.0.0.1:5500 내용:
이름을 입력하세요.





Javascript 자료형

자료형	설명
number	정수, 실수 등 산술 연산이 가능한 자료형
string	문자열로 이루어진 자료형
boolean	참/거짓으로 표현되는 논리 형태의 자료형
undefined	값이 할당된 적이 없거나 존재하지않는 속성일 때의 유형
null	아무런 값을 나타내지 않을 때의 자료형



Javascript 자료형

number형)

```
var num1 = 65;  
var num2 = 99.9;
```

boolean형)

```
var isChecked = true;  
var isPlaying = false;
```

string형)

```
var name = '홍0동';  
var fruit = "사과";
```

undefined&null형)

```
var data; //undefined  
var value = null;
```



Javascript 연산자

연산자	종류
산술연산자	+ - * / %
증감연산자	++ --
대입연산자	= += -= *= /= %=
연결연산자	+(문자열 결합)
비교연산자	== != ===(타입과 값을 같이 비교) !== > >= < <=
논리연산자	&& !
조건연산자	조건 ? 실행문1 : 실행문2;



== VS ===

10 == '10'
true

자동으로 자료형 변환
(동등연산자)

vs

10 === '10'
false

정확히 값과 자료형 비교
(일치연산자)



Javascript 연산자 실습

```
let num = 5;
```

```
let str = '5';
```

```
console.log(num == str);
```

```
// true
```

```
console.log(num != str);
```

```
// false
```

```
console.log(num === str);
```

```
// false
```

```
console.log(num !== str);
```

```
// true
```

```
console.log(num === Number(str));
```

```
// true
```

```
console.log(num !== Number(str));
```

```
// false
```

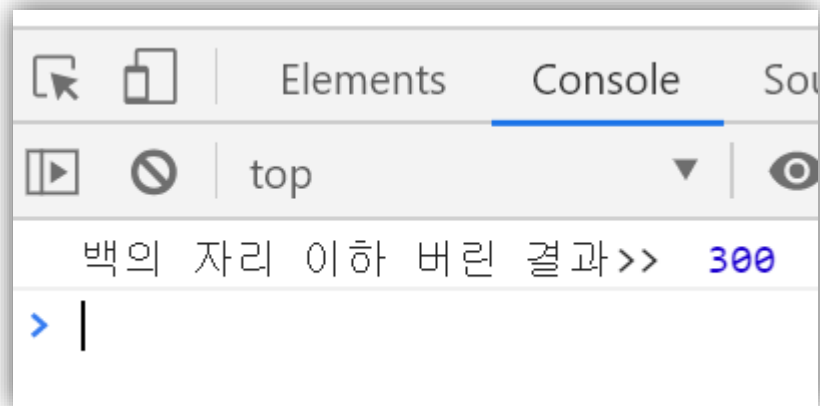



Javascript 연산자 실습

변수 num 값 중에서 백의 자리 이하를 버리는 코드이다.
만일 변수 num의 값이 456이라면 400이 되고, 111이라면 100이 된다.

```
let num = 312
```

?





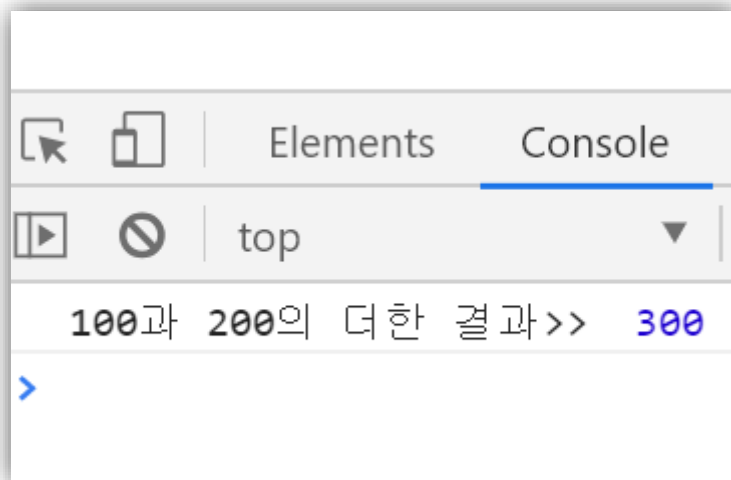
Javascript 형변환

연산자	종류
parseInt()	문자열을 숫자(정수)로 변경
parseFloat()	문자열을 숫자(실수)로 변경
Number()	다른 자료형을 숫자형(정수&실수)으로 변환
toString()	숫자를 문자열로 변경



Javascript 형변환

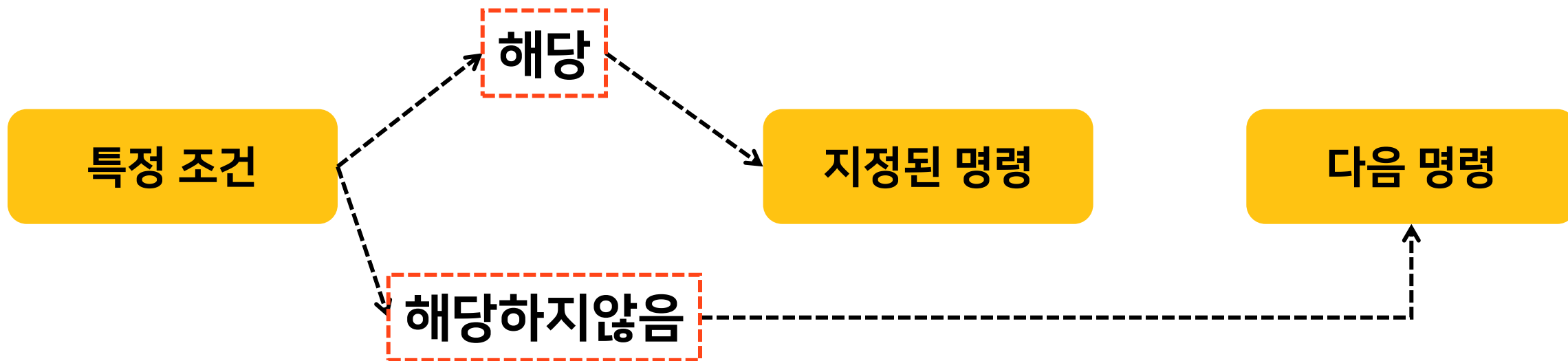
각각 값이 '100', 200인 num1, num2가 있다.
형변환을 하여 아래와 같은 결과를 출력하시오.





조건문

< 주어진 조건을 비교·판단하여 결과를 얻는 구문 >





조건문의 4가지 종류

단순 if문

if-else문

다중 if문

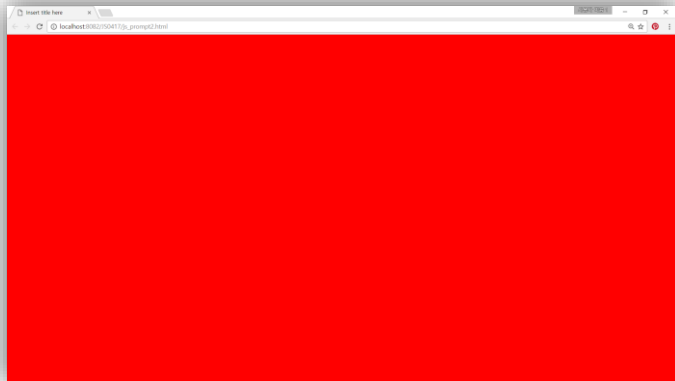
switch문



빨강, 초록, 파랑 중에 원하는 색을 입력했을 때, 웹 브라우저 배경색이 바뀌도록 작성하세요

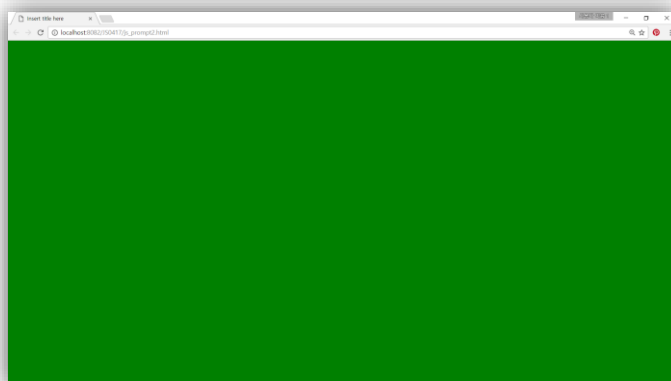
localhost:8082 내용: ×

색깔을 적어주세요(빨강,초록,파랑)

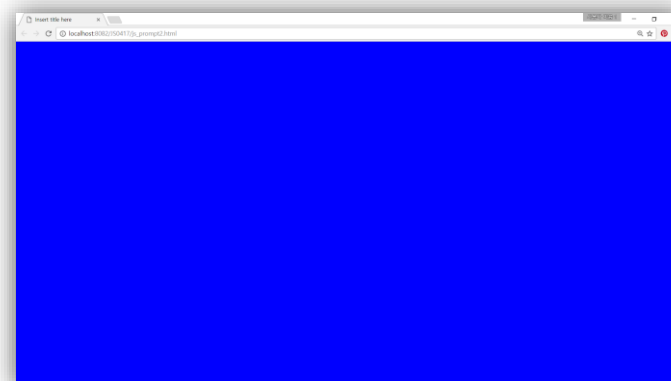
localhost:8082 내용: ×

색깔을 적어주세요(빨강,초록,파랑)

☐ 이 페이지가 추가적인 대화를 생성하지 않도록 차단합니다.
 

localhost:8082 내용: ×

색깔을 적어주세요(빨강,초록,파랑)

☐ 이 페이지가 추가적인 대화를 생성하지 않도록 차단합니다.
 



반복문

< 어떤 조건에 만족할 때까지 같은 처리를 반복하여 실행하는 구문 >

while문

do-while문

for문



조건문&반복문 실습

다음 랜덤하게 뽑인 숫자를 맞추는 프로그램입니다.
아래 결과와 같은 결과를 출력하시오.

127.0.0.1:5500 내용:
숫자를 입력해주세요!

1.입력

127.0.0.1:5500 내용:
입력한 숫자보다 작은 수 입니다...

127.0.0.1:5500 내용:
입력한 숫자보다 큰 수 입니다...

2.비교결과출력

127.0.0.1:5500 내용:
정답입니다 축하합니다 🎉

3.결과출력



Javascript 실행위치

내부

외부

인라인



<head> 혹은 </body> 태그 앞에 작성.

<html>

<head>

<meta charset = "EUC-KR">

<title>JavaScript</title>

</head>

<body>

</body>

</html>

<script type="text/JavaScript">
스크립트 실행문 작성
</script>



* **작성위치에 따라** 실행순서와 브라우저 렌더링에 영향을 미친다.

<head> 내 작성

단순한 로직처리(설정값 초기화)

```
<html>
  <head>
    <script>
      //javascript 코드작성
    </script>
  </head>
  <body>

  </body>
</html>
```

</body> 앞 작성

복잡한 로직처리(제어)

```
<html>
  <head>
  </head>
  <body>
    <script>
      //javascript 코드작성
    </script>

  </body>
</html>
```



외부 Javascript

```
<html>
<head>
  <script language="javascript" src="파일이름.js"></script>
  <meta charset = "EUC-KR">
  <title>JavaScript</title>
</head>
<body>

</body>
</html>
```

- 1.관리가 편리하다.
- 2.소스를 숨길 수 있다.
- 3.재사용할 수 있다.



HTML태그 내부에 이벤트 속성 삽입

```
<html>
<head></head>
<body>
  <input type="button" onclick="alert('Hello world')"
    value="버튼클릭!">
</body>
</html>
```

JS



{ 다음시간에 배울 내용 }
- 배열, 함수, 객체