

유노코딩과 함께하는 자바스크립트 기초 강의

강사 소개 & 강의 소개





강사 소개



강운호(유노코딩)

문과 감성 충만한 이과 출신 IT 강사입니다.
주로 강의나 교육 콘텐츠 만드는 일을 하고 있어요.
책, 고양이, 커피, 그리고 달리기를 좋아합니다 :)

- 2015~ 프로그램 개발자
- 2016~ IT 강사
- 2020~ 유튜브 채널 <유노코딩> 운영
- 2021~ IT 도서 저자
- 2022~ 직업 훈련 교사



이 강의에서는 무엇을 배우나요?

웹 페이지의 뼈대를 만드는 언어 HTML



웹 페이지의 스타일을 정의하는 언어 CSS



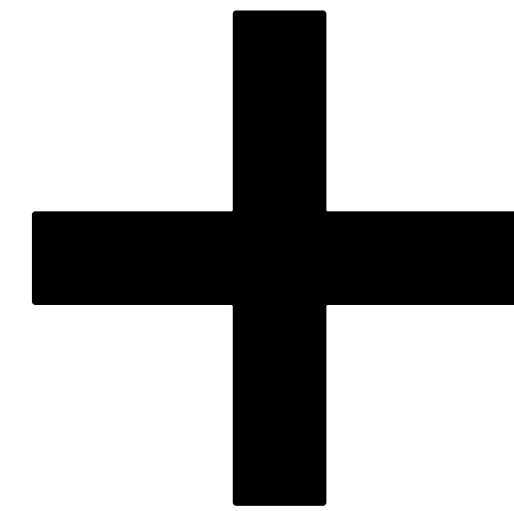


이 강의에서는 무엇을 배우나요?

웹 페이지의 뼈대를 만드는 언어 HTML



웹 페이지의 스타일을 정의하는 언어 CSS





이 강의에서는 무엇을 배우나요?

웹 페이지의 뼈대를 만드는 언어 HTML



웹 페이지의 스타일을 정의하는 언어 CSS



+

JS



강의 대상 & 내용 구성

이 강의는 초급 수준의 웹 개발자 위해 만든 자바스크립트 기초 강의입니다. 순수 입문자가 아닌, HTML&CSS의 역할을 이해하고 기본적인 사용법을 습득한 초급자를 대상으로 합니다.

이 강의는 자바스크립트를 활용하여 웹 페이지를 개발하는 방법을 교육하기 위해 제작된 강의입니다. 따라서 실습 시 HTML&CSS 기술이 많이 활용될 것이고, 그에 대한 부연 설명 또한 종종 진행될 예정입니다.



우리의 목표!

이 강의를 통해 저와 여러분이 이루고자 하는 목표는 다음과 같습니다!

- 웹 개발 분야에서 자바스크립트의 역할이 무엇인지 이해한다.
- 자바스크립트의 동작 방식과 문법을 이해하고 활용할 수 있다.
- 웹 페이지 만들기 실습을 통해 코딩에 재미를 붙이고, 자신감을 얻는다.





우리의 목표!

이 강의를 통해 저와 여러분이 이루고자 하는 목표는 다음과 같습니다!

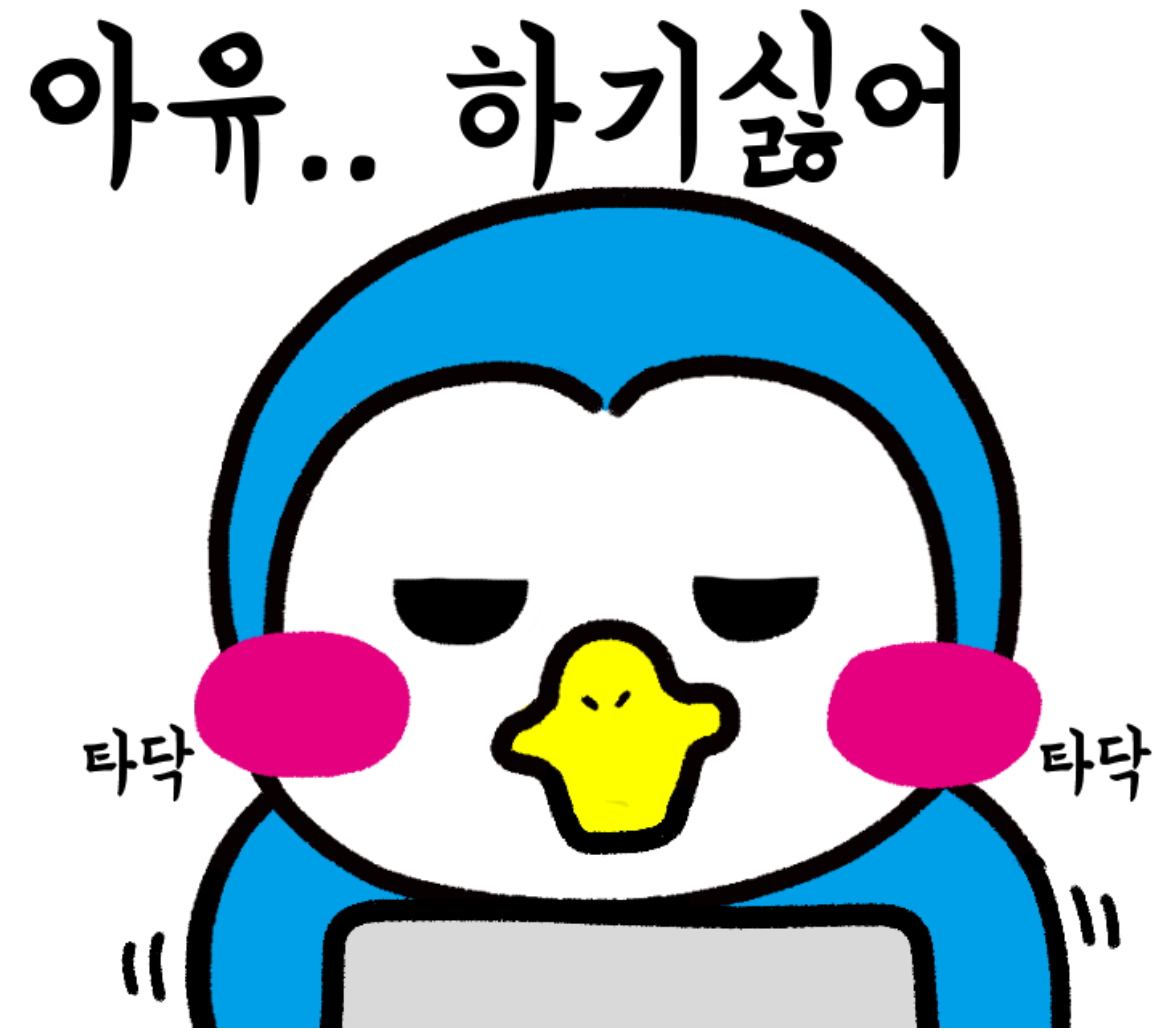
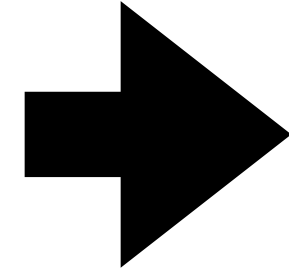
- 웹 개발 분야에서 자바스크립트의 역할이 무엇인지 이해한다.
- 자바스크립트의 동작 방식과 문법을 이해하고 활용할 수 있다.
- 웹 페이지 만들기 실습을 통해 코딩에 재미를 붙이고, 자신감을 얻는다.

재미있게 해야 늘어요!
즐기면서 합시당 ㅎㅎ

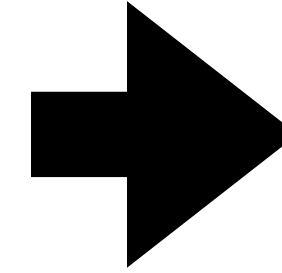
JS



기초 이론



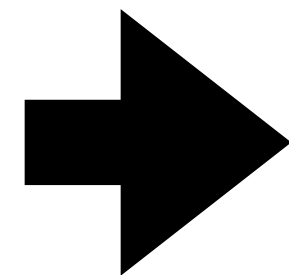
함께 실습



이해 완료(그러나 복습은 필수!)



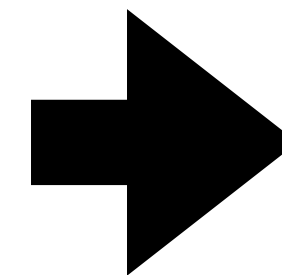
기초 이론



아유.. 하기싫어



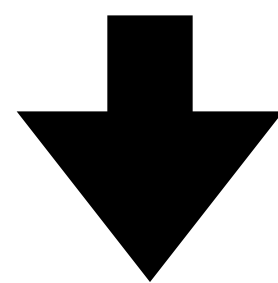
함께 실습



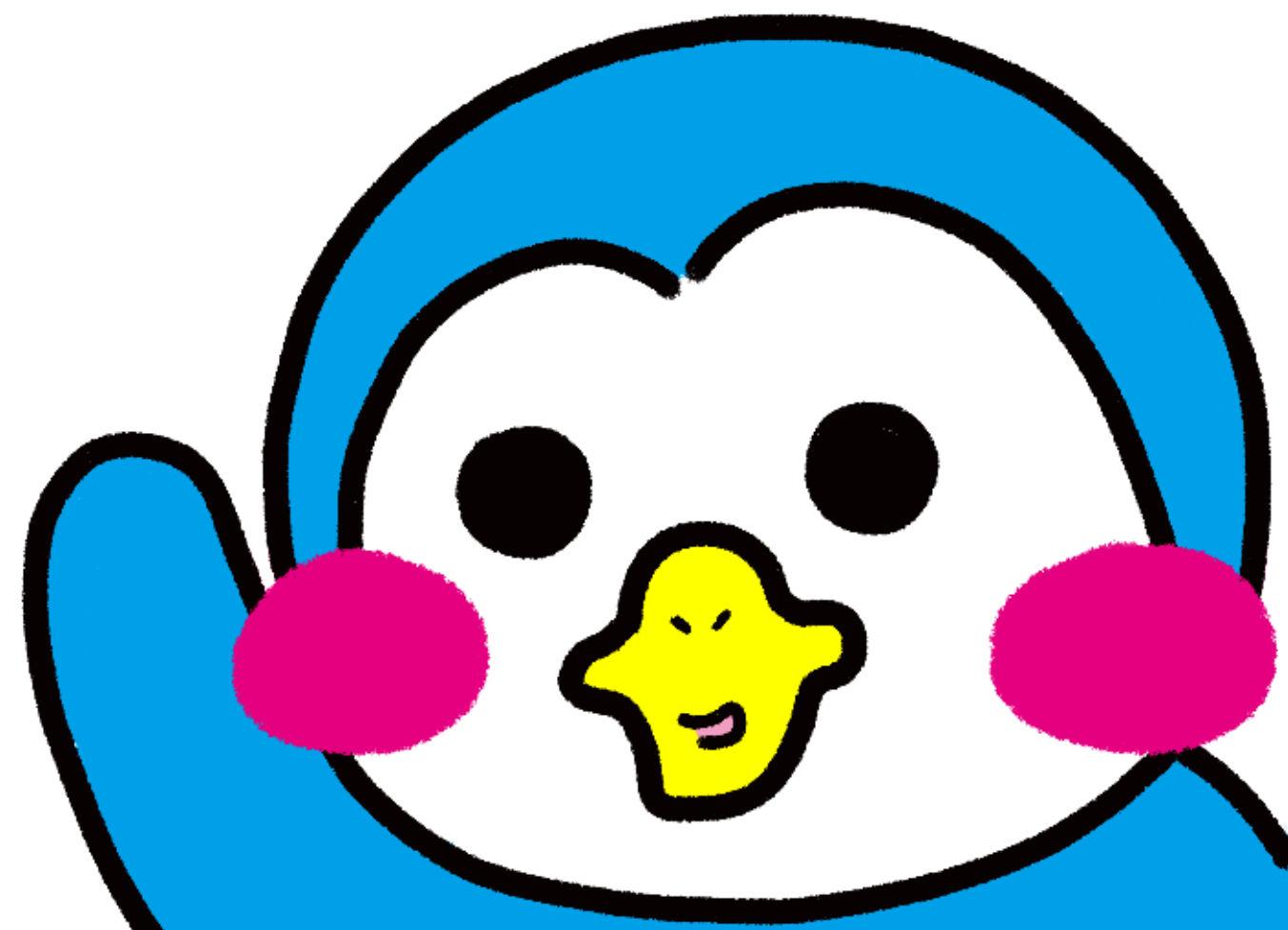
끝~



이해 완료(그러나 복습은 필수!)



웹 페이지 만들기 실습으로
실력과 자신감, 두 마리 토끼 잡기!

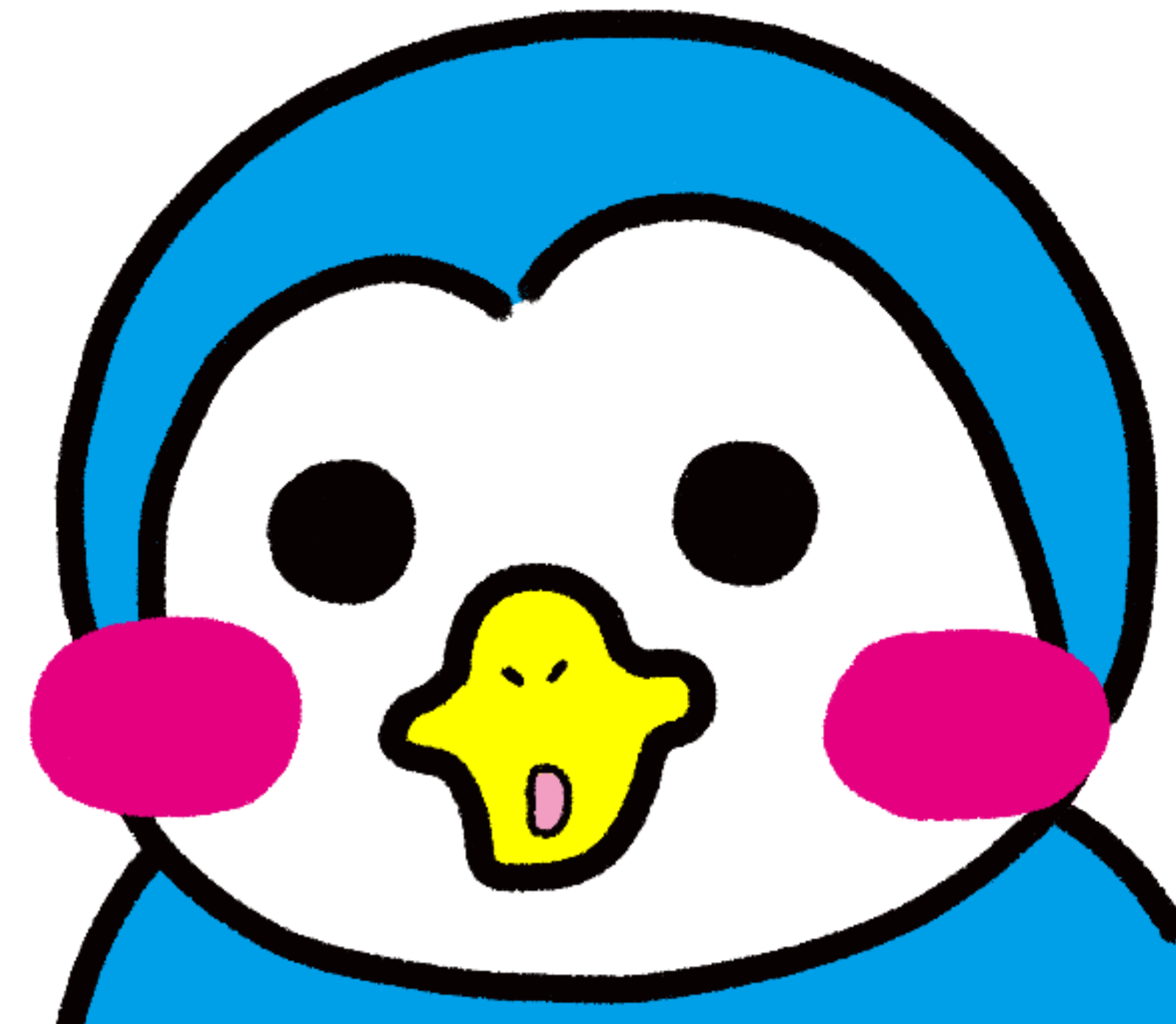




잘부탁드립니다 :)

유노코딩과 함께하는 자바스크립트 기초 강의

자바스크립트 소개





자바스크립트

HTML은 마크업 언어, CSS는 스타일시트 언어이다.
그렇다면 자바스크립트는? 자바스크립트는 프로그래밍 언어이다.

자바스크립트는 서버 개발, 어플리케이션 개발 등 다양한 목적을 위해 사용할 수 있는 팔방미인(?) 언어이지만, 주된 활동 분야는 역시 '웹 개발'이다!

웹 개발이란?

인터넷을 통해 서비스되는 웹사이트를 개발하는 일!



HTML

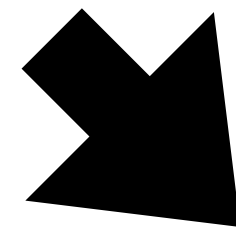
웹 콘텐츠를
정의한다



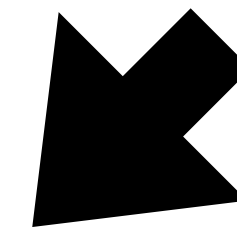
CSS

웹 콘텐츠를
스타일링한다

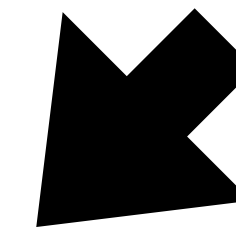
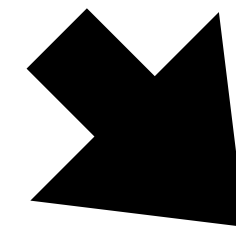
웹사이트 개발 삼형제



웹페이지는 이렇게 생겼어!



웹사이트 개발 삼형제



웹사이트 개발 삼형제



정리! 자바스크립트의 역할은?

웹사이트 개발에 있어, 자바스크립트의 역할은

- 웹브라우저 및 하위 객체가 가진 기능을 구동하거나
- HTML/CSS 를 통해 렌더링된 화면을 조작할 수 있다.

위와 같은 역할을 정상적으로 수행하기 위해서는,
자바스크립트 코드를 올바른 위치에 작성하는 것이 중요하다!



자바스크립트 작성 위치

웹 페이지에 자바스크립트 코드를 적용시키는 방법으로는 다음 두 가지가 있다.

- HTML문서 내부에 작성하기
- 자바스크립트 파일을 만들고, 그 안에 작성한 코드를 HTML 문서에 연결하기

내부에 작성할 땐 `<script>` 태그를 사용해요.
자바스크립트 파일을 별도로 만들 때는
확장자가 js인 파일을 만들어요.

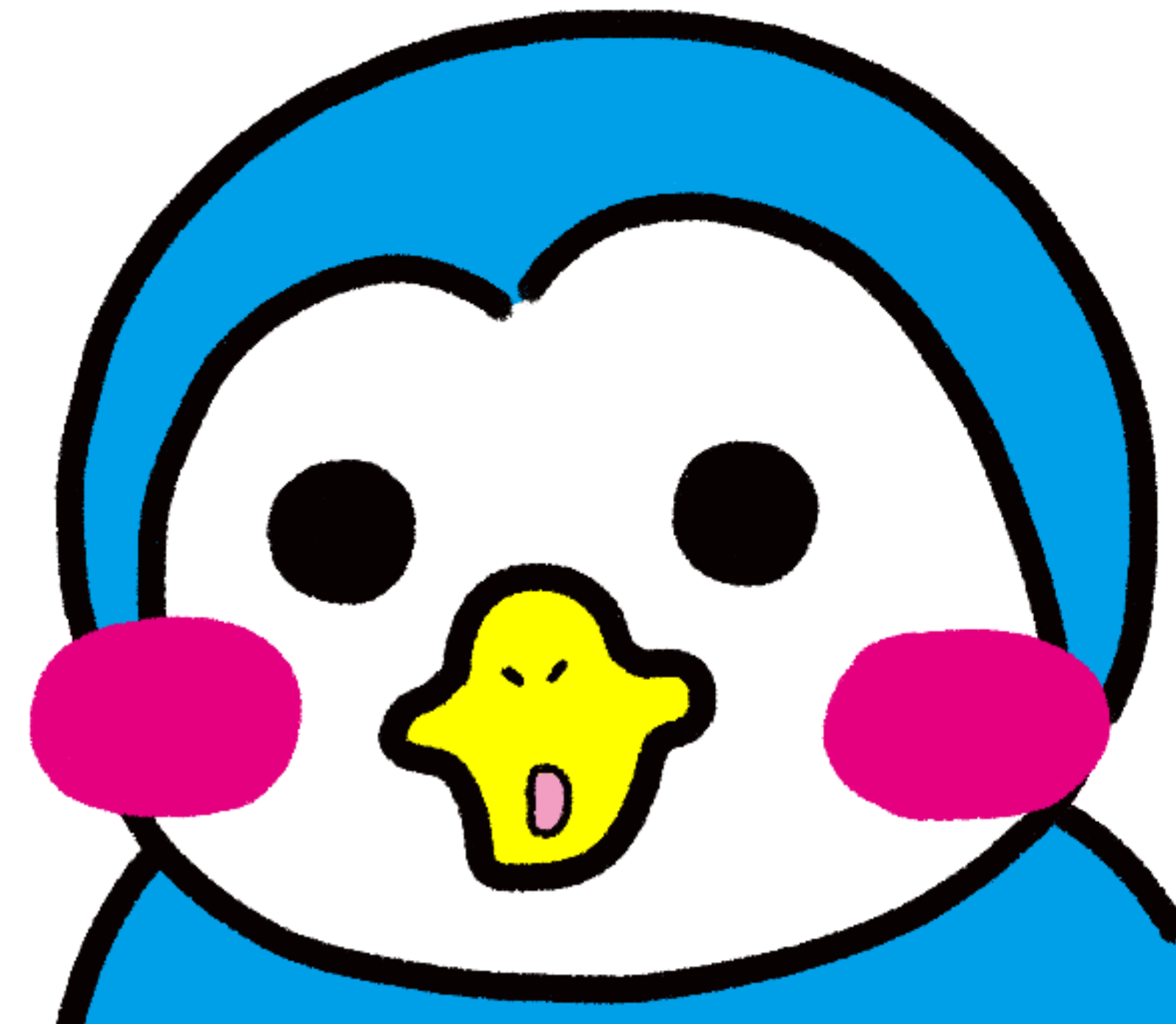




수고하셨습니다 :)

유노코딩과 함께하는 자바스크립트 기초 강의

자바스크립트 동작원리
& 출력해보기





스크립트 동작 원리

자바스크립트는 프로그래밍 언어로써, 소프트웨어적 대상에게 명령을 내리는 역할을 수행한다. 여기에서 소프트웨어적 대상이란, 웹브라우저, 웹 요소, 웹 스타일 등을 의미한다.

명령을 내리고픈 대상에 자바스크립트 명령을 전달하면, 다음 절차대로 작업이 이루어진다.



자바스크립트 명령문으로 어떤 처리를 지시하고자 하는 대상을 가리켜 ‘객체’라 한다.



입력, 처리, 출력

입력, 처리, 출력은 각각 다음과 같은 의미를 가진다.

- 입력 : 객체에게 어떤 작업을 수행하라는 명령을 전달한다.
- 처리 : 객체가 주어진 작업을 수행한다.
- 출력 : 객체가 작업을 수행한 결과를 사용자에게 반환한다.

자바스크립트의 모든 명령문은 대~부분 위와 같은 절차에 따라 수행된다!

그리고 자바스크립트에는 다양한 객체 이름이 미리 정의되어 있어 사용자는 이를 어떻게 사용하는지만 알고 있으면 된다.



객체 다루는 방법

자바스크립트가 제공하는 객체 이름을 선택하고, 뒤에 점(.)을 붙인 다음 전달하고자 하는 명령을 적어준다. 그러면 입력, 처리, 출력의 절차가 시작된다.

객체 명령의 두 가지 유형

객체.데이터

객체가 가지고 있는 숫자, 문자 등의 다양한 데이터를 사용할 수 있다
객체가 가지고 있는 데이터(정보)를 가리켜 일반적으로 '속성'이라 한다

객체.기능()

객체가 가지고 있는 다양한 기능을 수행할 수 있다(괄호 필수).
객체가 가지고 있는 기능을 가리켜 일반적으로 '메소드'라 한다



모호하게 느껴지나요? 그렇다면 구체적인 예 한 가지

웹브라우저는 소프트웨어 세계에 존재하는 사물, 즉 객체이다. 그리고 자바스크립트에는 웹브라우저에 명령을 전달할 수 있는 객체 이름이 정의되어 있다.

나를 이용하면
웹브라우저에게 다양한
명령을 내릴 수 있어!

window



`window.alert(x)`

`window.alert(x)` 명령문은 웹브라우저에게 ‘경고창에 x라고 쓴 다음 화면에 띄워줘!’라는 명령을 전달한다. x에는 사용자가 원하는 내용을 입력하면 된다.

‘warn’이라고 쓰여진
경고창이 나타나게 돼요!

`window.alert('warn')`



해보기 전에 잠깐! 주석

HTML&CSS와 마찬가지로 자바스크립트 코드에서도 주석을 사용할 수 있다.
두 가지 방법을 사용할 수 있다.

// 주석이다



주석을 한 줄밖에 쓸 수 없다

/* 주석이다 */



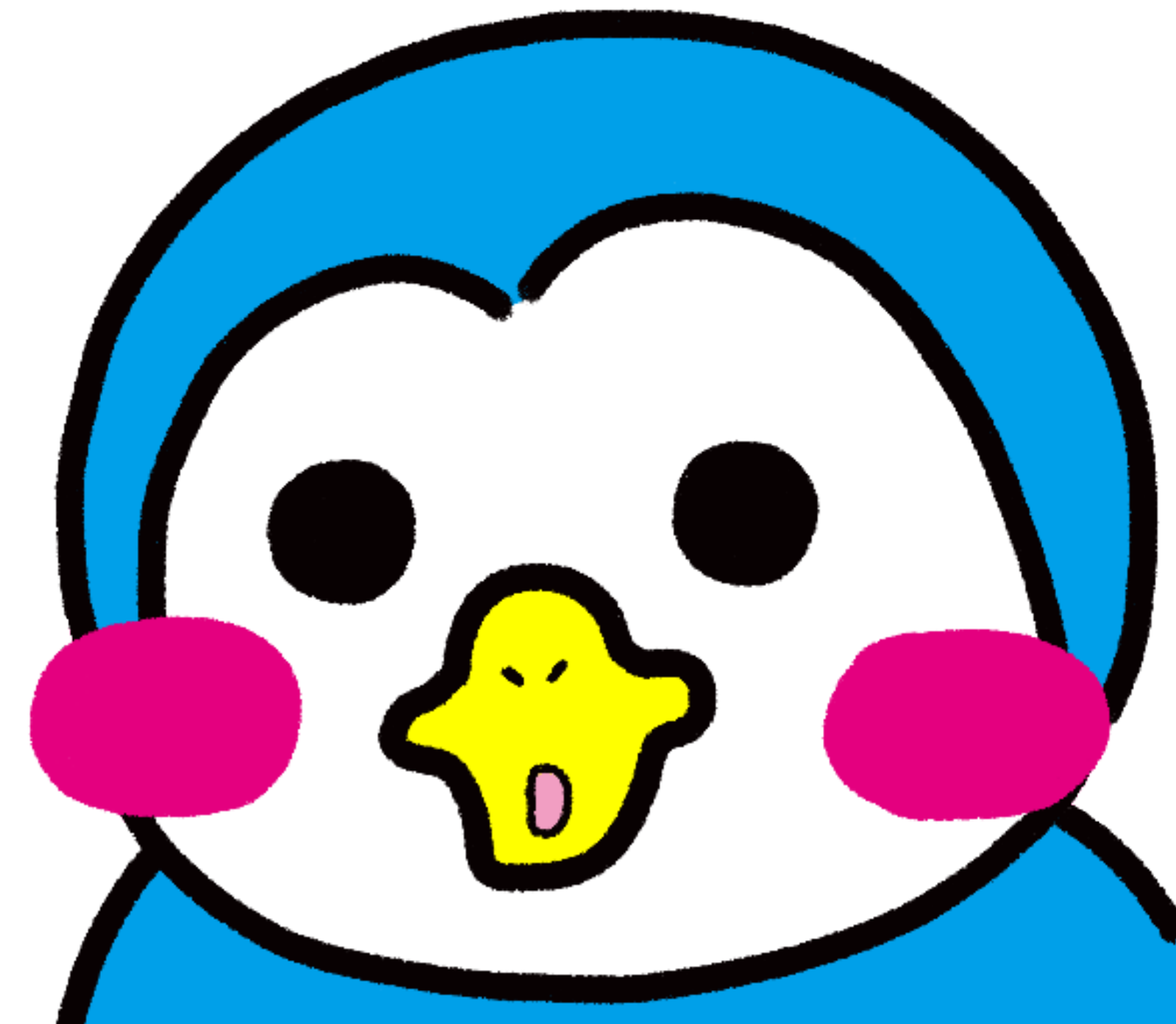
주석을 여러 줄에 걸쳐 쓸 수 있다



수고하셨습니다 :))

유노코딩과 함께하는 자바스크립트 기초 강의

콘솔 출력 &
자료형 개념 소개





콘솔을 소개합니다

콘솔(console)은 브라우저의 디버깅 콘솔을 의미한다. 콘솔은 브라우저 안에 내장된 브라우저의 하위 객체이므로 브라우저 객체를 통해 접근할 수 있다.

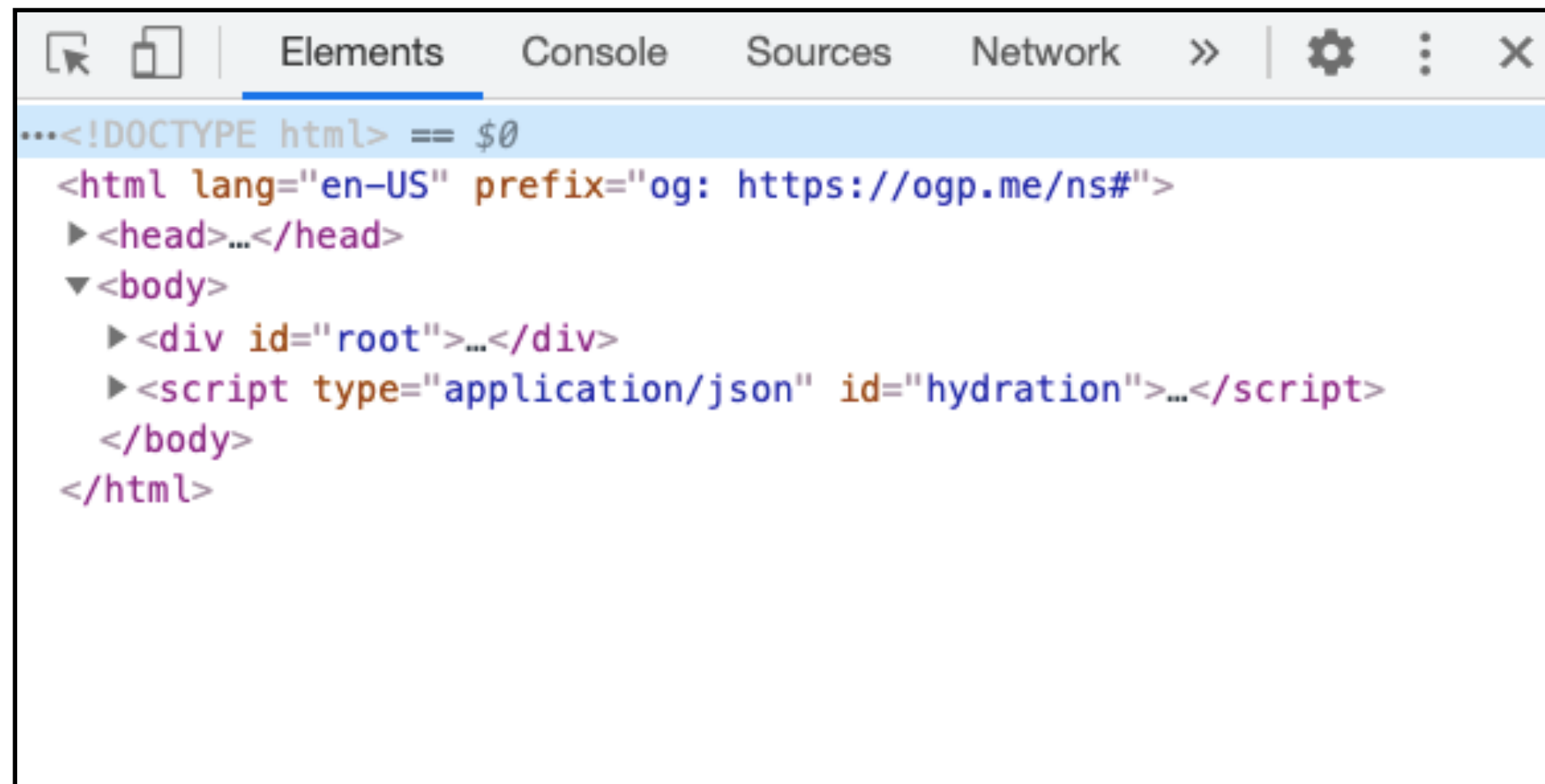
객체 뒤에 . 을 찍으면
객체가 가지고 있는 데이터에
접근할 수 있다

window.console

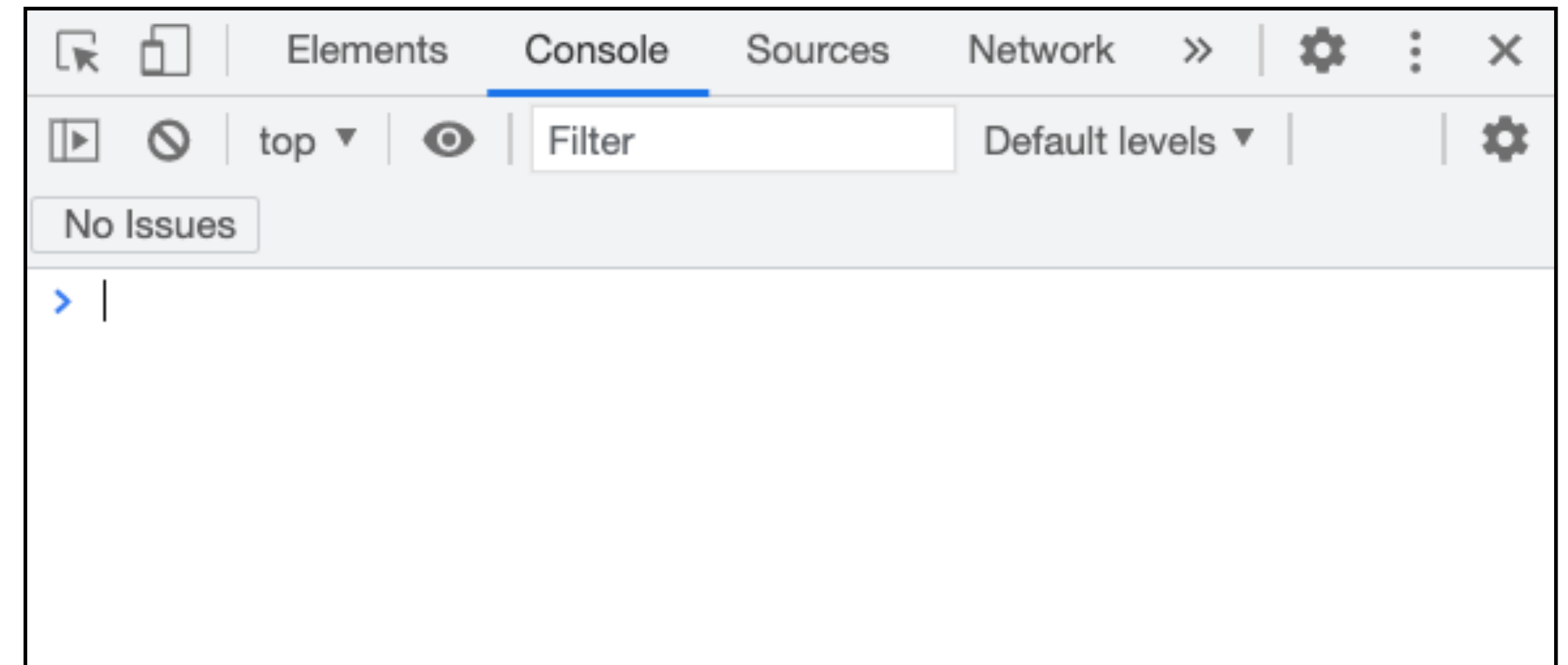
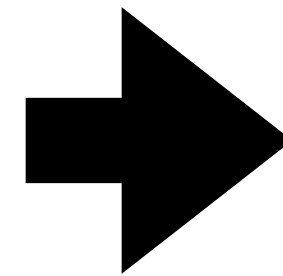


디버깅 콘솔이 뭔데

웹브라우저의 ‘개발자 도구’를 열면 메뉴 중 ‘콘솔(console)’이라는 항목이 존재하는데, 이것을 누르면 나타나는 화면이 바로 디버깅 콘솔이다.



```
...<!DOCTYPE html> == $0
<html lang="en-US" prefix="og: https://ogp.me/ns#">
  ><head>...</head>
  ><body>
    ><div id="root">...</div>
    ><script type="application/json" id="hydration">...</script>
  </body>
</html>
```



=> 디버깅 콘솔을 이용하면 자바스크립트 코드를 테스트(점검)할 수 있다!



별 거 아니지만 꿀팁

우리가 웹브라우저에 접근할 수 있게 해주는 객체 window는 console을 비롯한 다양한 하위 객체 및 데이터, 메소드 등을 포함하고 있는데, 사용자는 편의를 위해 window.을 생략한 채로 코드를 작성할 수 있다.

window.console



이렇게 쓰나

console



요렇게 쓰나 똑같음



console.log(x) & 자료형

console.log(x) 명령문은 디버깅 콘솔에게 ‘콘솔창에 x라고 써 줘!’ 라는 명령을 전달한다. x 자리에는 사용자가 원하는 내용을 입력하면 되는데, 자바스크립트 문법으로 표현될 수 있는 데이터만이 x 자리에 들어갈 수 있다.

자바스크립트에는 데이터를 표현할 때 사용하는 몇 가지 형식이 정의되어 있는데, 이러한 형식을 통틀어 ‘자료형’이라 한다!

console.log(x)



강의를 통해 살펴 볼 자료형

자바스크립트 주요 자료형으로는 다음과 같은 것들이 있다.

- 숫자
- 문자열
- 불리언
- 다양한 목적과 형태의 객체

=> 이 시간엔 숫자에 대해서 알아보고, 다른 건 강의를 진행하면서 차차 알아갈게요!



자바스크립트에서 숫자를 사용할 때는

숫자는 우리가 알고 있는 그 숫자가 맞다. 숫자는 정수와 실수로 구분할 수 있다.

정수를 콘솔에 출력하기

```
console.log(88)  
console.log(0)  
console.log(-8)
```

실수를 콘솔에 출력하기

```
console.log(1.8)  
console.log(-8.1)
```




숫자인데 산술 연산 정도는 해 줘야

산술 연산은 더하기, 빼기 등 수를 이용한 계산이다. 두 개의 숫자 데이터를 피연산자로 받아서 하나의 숫자 데이터를 결과로 반환한다.

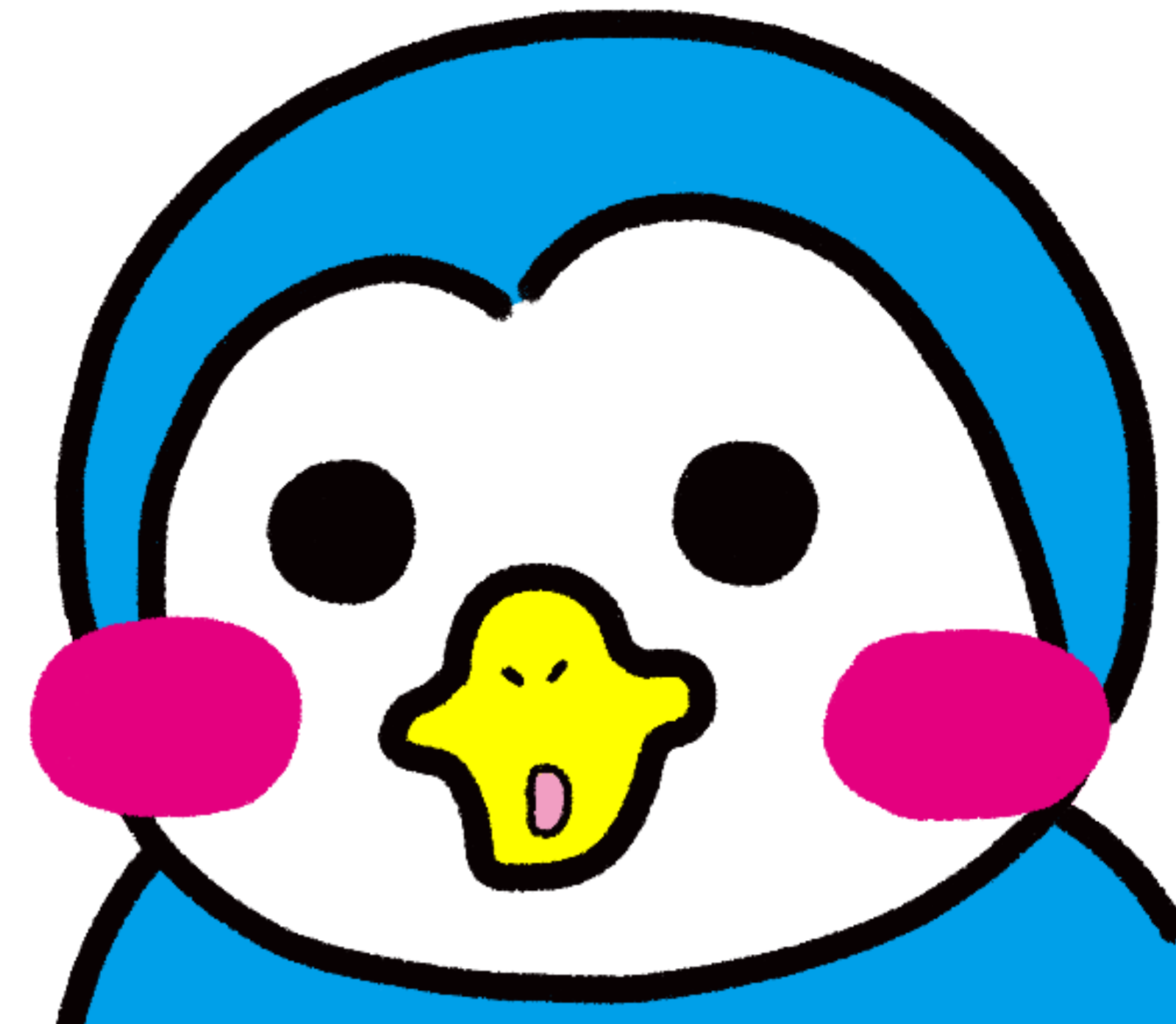
의미	연산자	예제
더하기	+	$2 + 2$ 는 4를 반환
빼기	-	$5 - 2$ 는 3을 반환
곱하기	*	$3 * 2$ 는 6을 반환
나누기	/	$1 / 2$ 는 0.5를 반환
나머지	%	$12 \% 5$ 는 2를 반환



수고하셨습니다 :)

유노코딩과 함께하는 자바스크립트 기초 강의

변수와 상수
그리고 대입





정말 중요한 개념, 변수!

변수란 '데이터에 붙이는 이름표'이다. 변수를 이용하면 이름표를 붙여 둔 데이터를 기억해두었다가 필요할 때마다 재사용 할 수 있게 된다.

변수 만드는 방법

```
let 변수이름 = 데이터;
```

이게 포인트!

변수가 생성되고 나면, 변수(변수이름)를 지정된 데이터 대신 사용할 수 있다.



변수의 선언 및 초기화

변수를 만드는 작업을 ‘변수의 선언’이라 부르고, 만들어진 변수에 첫 데이터를 지정하는 작업을 ‘변수의 초기화’라 부른다. 두 과정은 동시에 진행할 수도, 따로 진행할 수도 있다.

변수의 선언과 초기화 따로 하기

```
let 변수이름; // 변수 선언  
변수이름 = 데이터; // 변수 초기화
```

키워드 `let` 은 ‘변수를 만들겠다’라는 의미로 쓰인다.



변수는 변한다

변수는 데이터를 기억하기 위해 사용한다. 변수는 한 번에 하나의 데이터만 기억할 수 있으며, 기억하고 있는 값을 바꿔가며 사용할 수 있다.

변수가 기억하고 있는 데이터를 바꿀 수 있다

```
let 변수이름 = 데이터1; // 변수 선언과 초기화 동시 진행  
변수이름 = 데이터2; // 새로운 데이터 대입  
변수이름 = 데이터3; // 새로운 데이터 또 대입
```




변수의 이름은 개발자 맘대로. 그러나...

변수는 '데이터에 붙이는 이름표'이다(라고 했죠?). 변수의 이름은 기본적으로 짓는 사람 마음대로 지어줄 수 있지만, 몇 가지 제약사항도 존재한다.

제약사항

- 변수명에는 오직 문자와 숫자, 그리고 기호 \$과 _ 만이 포함될 수 있다.
- 변수명의 첫 번째 글자로 숫자가 올 수 없다.
- 이미 다른 뜻을 가지고 있는 단어(키워드)는 변수명으로 사용할 수 없다.



변수에는 값을 대입합니다

대입이란, 오른쪽 피연산자의 데이터를 왼쪽 피연산자에 집어넣는(?) 작업을 뜻한다.
변수에 값을 기억시키는 작업 또한 대입이며, 이때 대입 연산자를 사용한다.

의미	예제	예제 풀이
대입하기	$a = b$	값 b 를 변수 a 에 대입
더하여 대입	$a += b$	변수 a 의 값에 b 를 더한 결과를 a 에 대입
빼서 대입	$a -= b$	변수 a 의 값에서 b 를 뺀 결과를 a 에 대입
곱하여 대입	$a *= b$	변수 a 의 값에 b 를 곱한 결과를 a 에 대입
나누어 대입	$a /= b$	변수 a 의 값을 b 로 나눈 결과를 a 에 대입



잠깐! 변수 말고 상수도 있다

상수란 수식에서 ‘변하지 않는 값’을 뜻한다. 변수와 반대되는 것으로, 상수는 ‘단 하나의 데이터만을 위해 사용하는 이름표’로써, 값의 변경이 불가하다.

상수 만드는 방법

```
const 상수이름 = 데이터;
```

상수도 변수처럼 지정된 데이터 대신 사용할 수 있다.
상수 이름 짓기 규칙은 변수 이름 짓기와 동일하다.



상수의 선언과 초기화

상수를 만들 때는 선언과 동시에 초기화를 해주어야 한다. 그렇지 않으면 데이터를 지정할 수 없다.

상수의 선언과 초기화 따로 하면?

```
const 상수이름;  
상수이름 = 데이터; // 에러 발생!
```

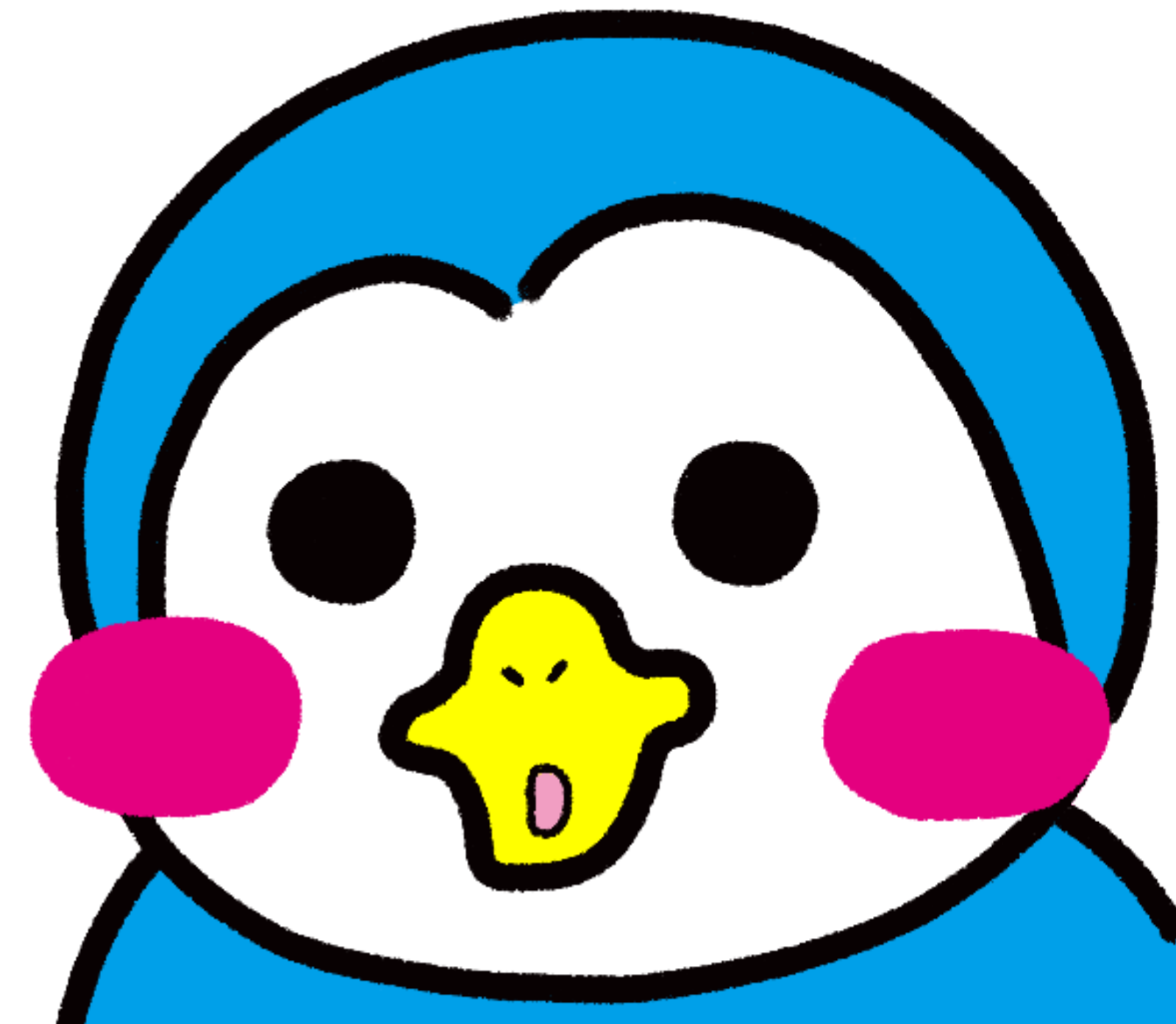
상수는 자신이 선언될 때 지정된 데이터로 고정된다. 선언 이후 데이터를 대입하려고 하면 에러 발생한다.



수고하셨습니다 :))

유노코딩과 함께하는 자바스크립트 기초 강의

문자열 &
`window.prompt()`





기본 자료형 문자열

자바스크립트에서 문자열이란, '기호의 순차 수열'을 뜻한다. 문자, 숫자, 특수문자 등 다양한 기호를 조합해 만들 수 있는 '기호의 집합'이며, 따옴표로 시작해 따옴표로 끝나는 데이터이다.

작은 따옴표, 큰 따옴표 모두 사용 가능

```
let string1 = '문자열에는 기호가 들어간다'  
let string2 = "12345...!?"
```

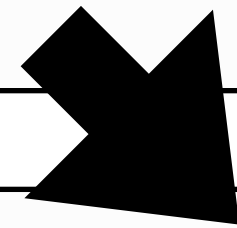
단, 시작하는 따옴표와 끝나는 따옴표가 같아야 한다!



문자열도 연산이 가능해요

자바스크립트 문자열을 대상으로 더하기(+) 연산을 수행할 수 있다.
이때 더하기는 산술적인 덧셈을 수행하지 않고, 문자열 이어붙이기를 수행한다.

"버즈가 부릅니다. " + "겁쟁이"



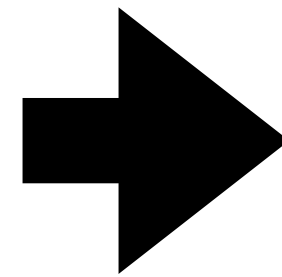
"버즈가 부릅니다. 겁쟁이"



window.prompt(x)

window 객체의 prompt(x) 메소드는 사용자로부터 문자열을 입력받을 수 있는 다이얼로그 박스를 열어주는 메소드이다. 괄호 안에 x로는 다이얼로그 박스에 띄울 메시지(문자열)를 전달할 수 있다.

```
prompt("다이얼로그 박스에 이렇게 쓸 거야");
```



이 페이지 내용:

다이얼로그 박스에 이렇게 쓸 거야

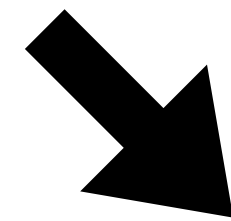
취소 확인



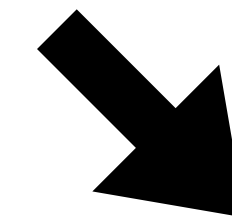
prompt(x)의 문자열 반환

prompt(x) 메소드는 실행 시에 사용로부터 문자열을 입력받고, 입력받은 문자열을 자신이 사용된 곳에 그대로 '반환'한다. 즉, 메소드가 문자열로 바뀌게 된다.

```
const data = prompt("아무거나 써 봐요!");
```



주의! 실제로 코드가 바뀌는 게 아닌
마치 바뀐 것처럼 실행된다는 의미이다!



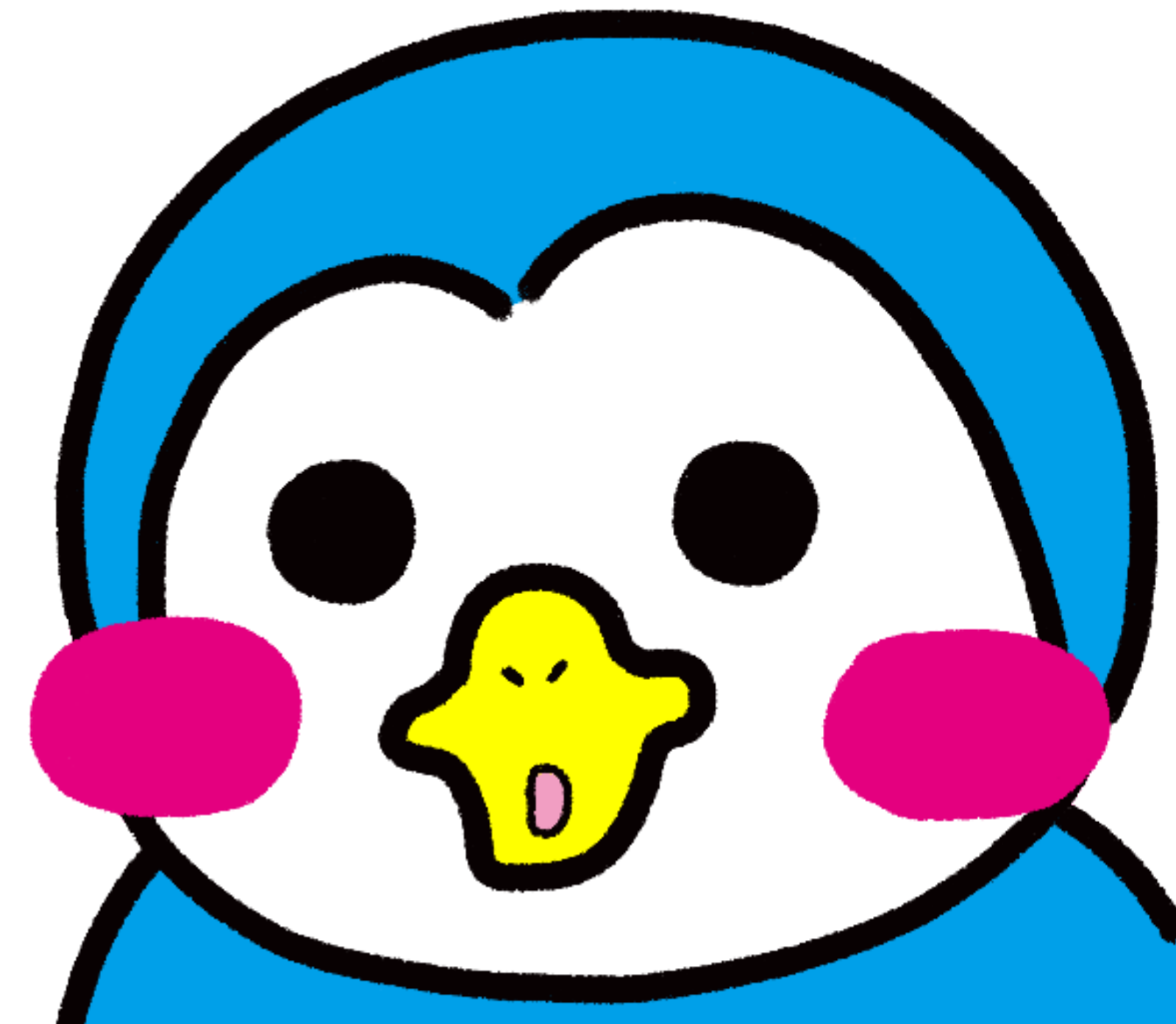
```
const data = "아무거나 썼다 어쩔래!";
```



수고하셨습니다 :))

유노코딩과 함께하는 자바스크립트 기초 강의

템플릿 리터럴





템플릿 리터럴

템플릿 리터럴은 문자열을 표현하는 또 다른 방법으로, 따옴표를 이용해 표현하는 방법에 비해 조금 늦게 추가된 문법이다. 템플릿 리터럴은 백틱을 이용해 표현한다. 백틱(`)은 키보드 왼쪽 상단에 있으며, 쉬프트 키와 함께 누르면 물결(~)이 된다.

```
// 따옴표를 이용한 기존의 문자열
```

```
const str1 = '작은 따옴표'
```

```
const str2 = "큰 따옴표"
```

```
// 백틱을 이용한 템플릿 리터럴
```

```
const str3 = `이게 백팁입니다`
```

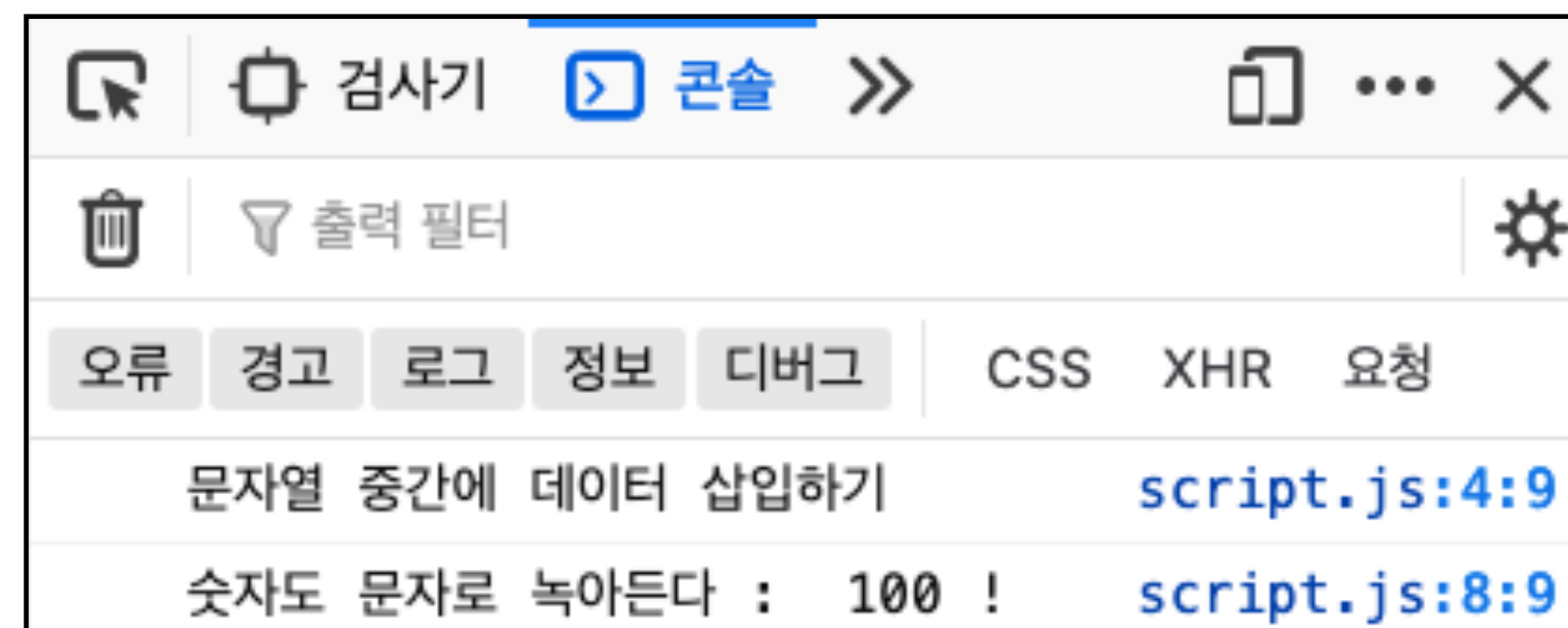
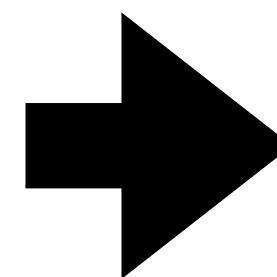


표현식 삽입 방법

템플릿 리터럴은 반환값이 존재하는 자바스크립트 코드, 즉 표현식을 내장할 수 있는 문자열 표현법이다. 이는 문자열의 내용에 데이터를 삽입한다는 것을 의미한다. 템플릿 리터럴로 표현한 문자열 내부에 플레이스홀더(`${}`)를 기입하고, 그 안에 데이터를 기입하면 데이터는 문자열의 멤버가 된다.

```
const data1 = "데이터"
const str1 = `문자열 중간에 ${data1} 삽입하기`
console.log(str1)

const data2 = 100
const str2 = `숫자도 문자로 녹아든다 : ${data2} !`
console.log(str2)
```

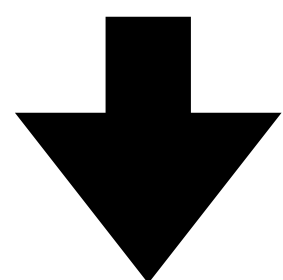




표현식 삽입 방법

데이터를 반환하는 메소드 또한 데이터와 같이 취급할 수 있다.

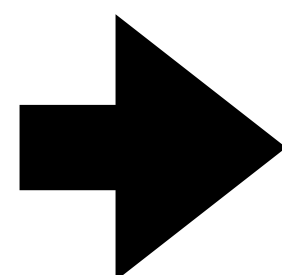
```
const str = `그렇다면 이건 될까? ${prompt('그렇다면 이건 될까?')}`  
console.log(str)
```



file://

그렇다면 이건 될까?

취소 확인



검사기 콘솔 >>

출력 필터

오류 경고 로그 정보 디버그 CSS XHR 요청

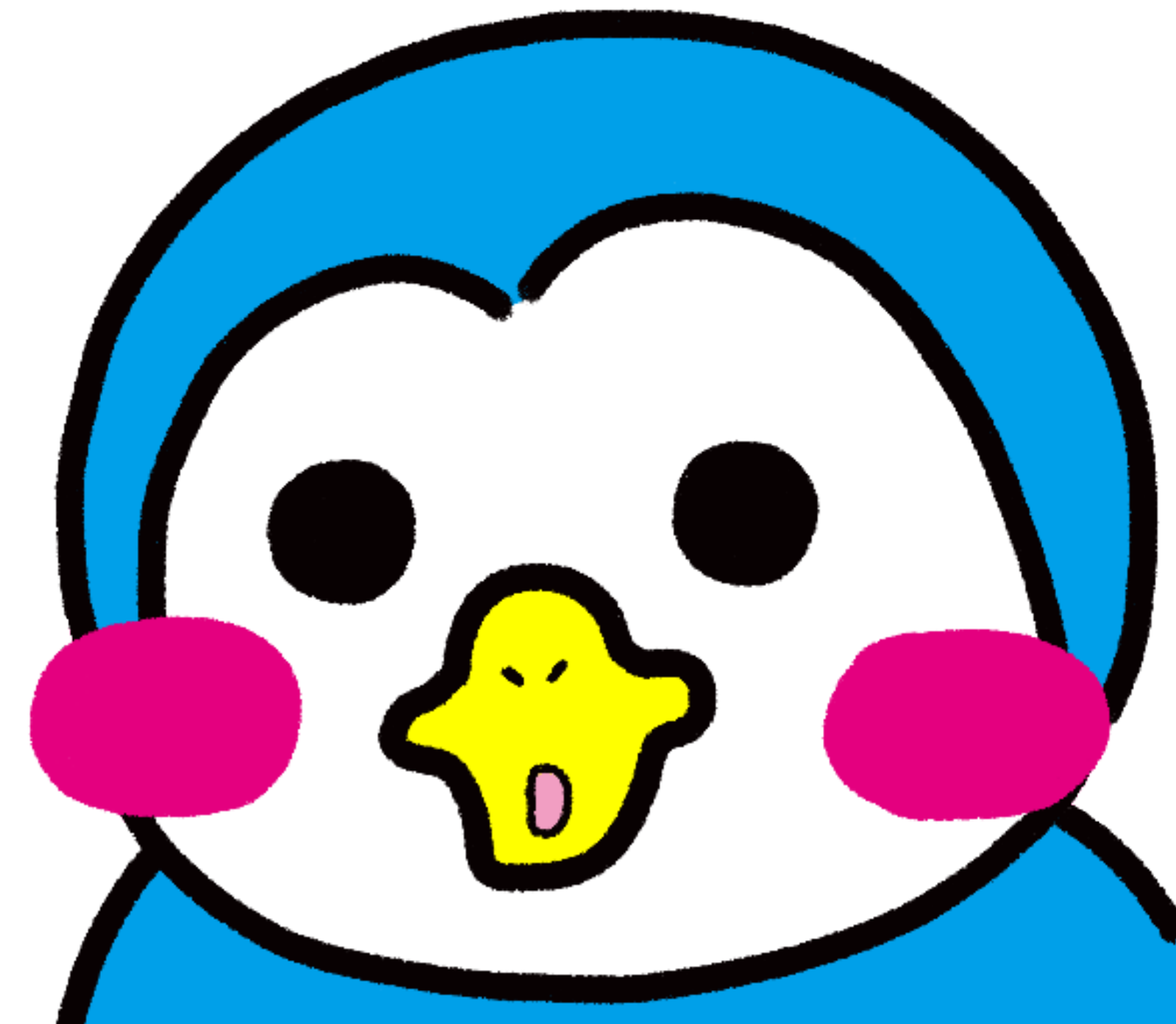
그렇다면 이건 될까? 한번 해보죠 뭐! script.js:4:9



수고하셨습니다 :)

유노코딩과 함께하는 자바스크립트 기초 강의

undefined, null, boolean
그리고 window.confirm()





undefined & null

자바스크립트 프로그래밍을 하다보면 undefined나 null이라는 키워드를 목격하는 경우가 생긴다. 언디파인드(undefined)는 아직 데이터가 정의되지 않았음을 나타내고, 널(null)은 의도적으로 데이터가 없음을 나타내기 위해 사용하는 일종의 표현 수단이다.

```
let number;  
  
// 여기에는 아직 값이 정의되지 않았어! undefined야.  
console.log(number); // undefined  
  
// number는 3이라고 정의한 다음 확인하자.  
number = 3;  
console.log(number); // 3
```

```
let number;  
  
// 이 변수는 숫자 0 이다!  
number = 0;  
  
// 이 변수에는 아무 것도 없다!  
number = null;
```




기본 자료형 boolean

불리언(불린이라고 해도 돼요)은 숫자, 문자열과 같은 데이터 타입 중 하나이다. true와 false, 단 두 가지 값만 존재한다. 참과 거짓 여부를 나타내기 위해 사용하는 데이터이다.

```
let value;  
value = true; // 소문자로 써야 합니다  
value = false; // 참, 거짓 모두 마찬가지로  
value = False; // 이거 안되고  
value = True; // 이것도 안 돼요
```



`window.confirm(x)`

`window.confirm(x)`는 사용자에게 확인과 취소 둘 중 하나를 선택하게 하고, 선택에 따라 `true` 또는 `false`를 반환하는 명령이다.

chrome://new-tab-page 내용:

사실입니까?

취소

확인

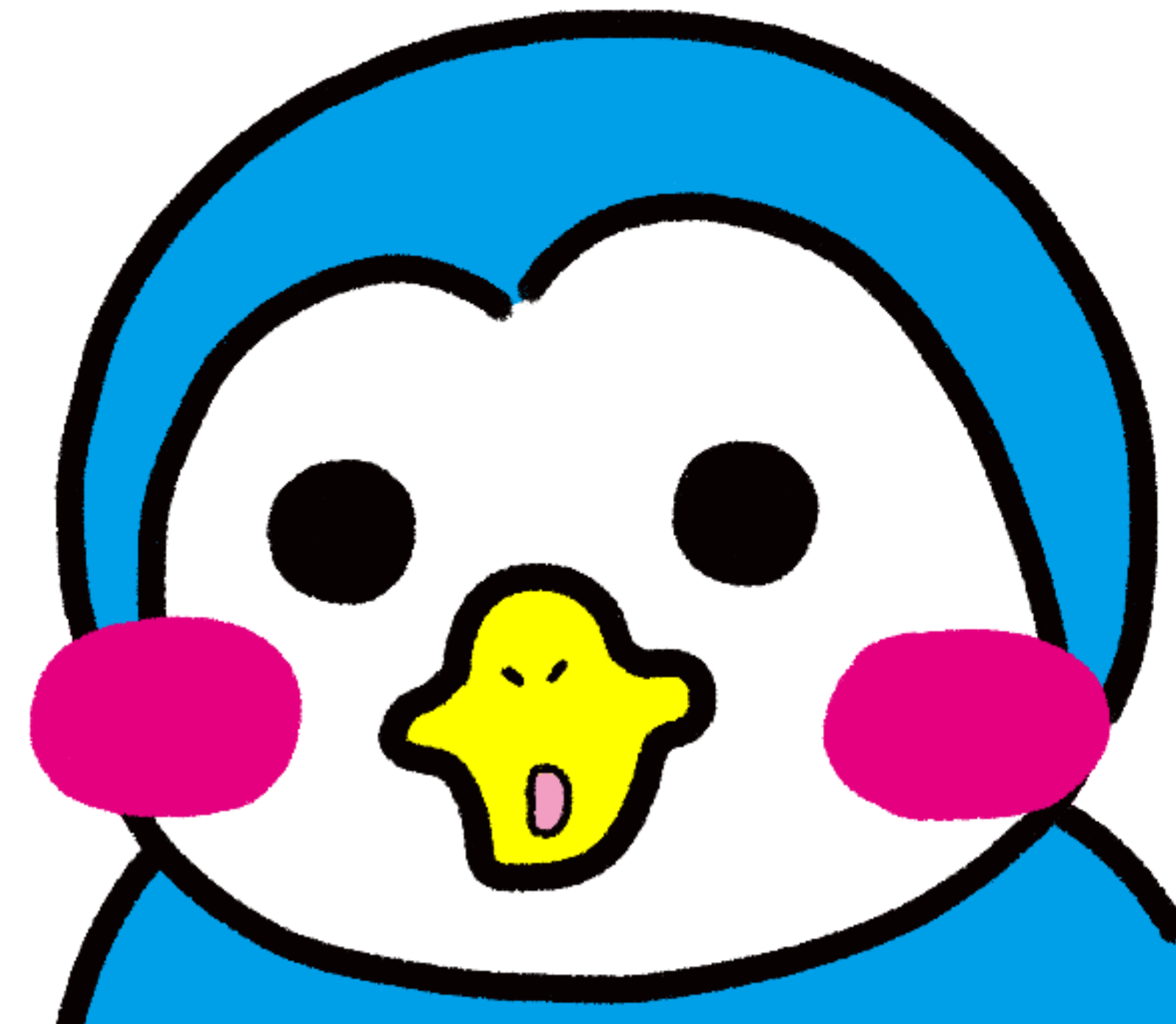
=> 당연히, `x`에는 '사실입니까?'가 입력되었다!



수고하셨습니다 :))

유노코딩과 함께하는 자바스크립트 기초 강의

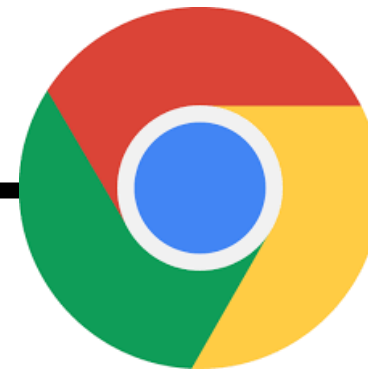
DOM 소개 &
콘텐츠 변경하기





이 정도는 알고 계시죠?!

웹브라우저는 HTML 문서를 해석하고, 화면을 통해 해석된 결과를 보여준다.
해석한 HTML 코드를 화면을 통해 보여주는 과정을 '렌더링'이라 한다.



코드 해석

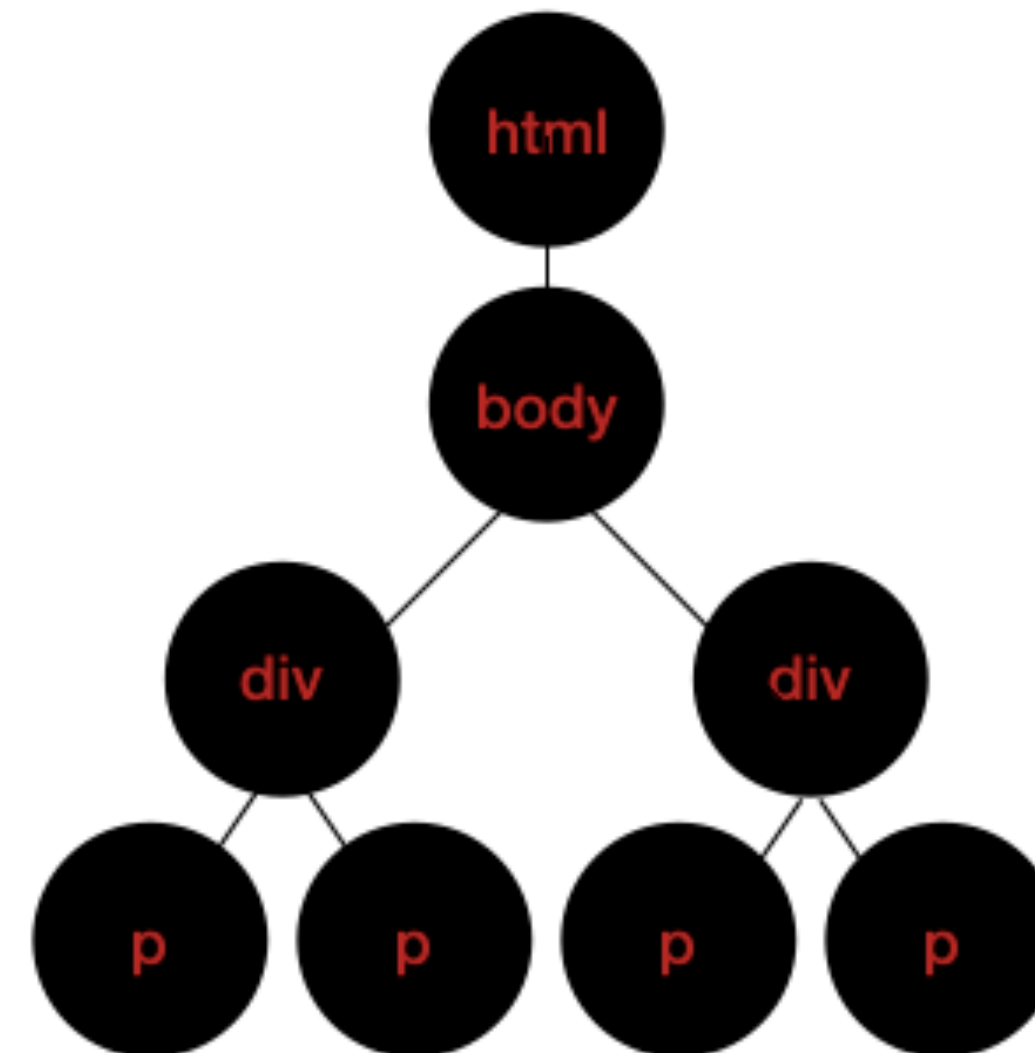


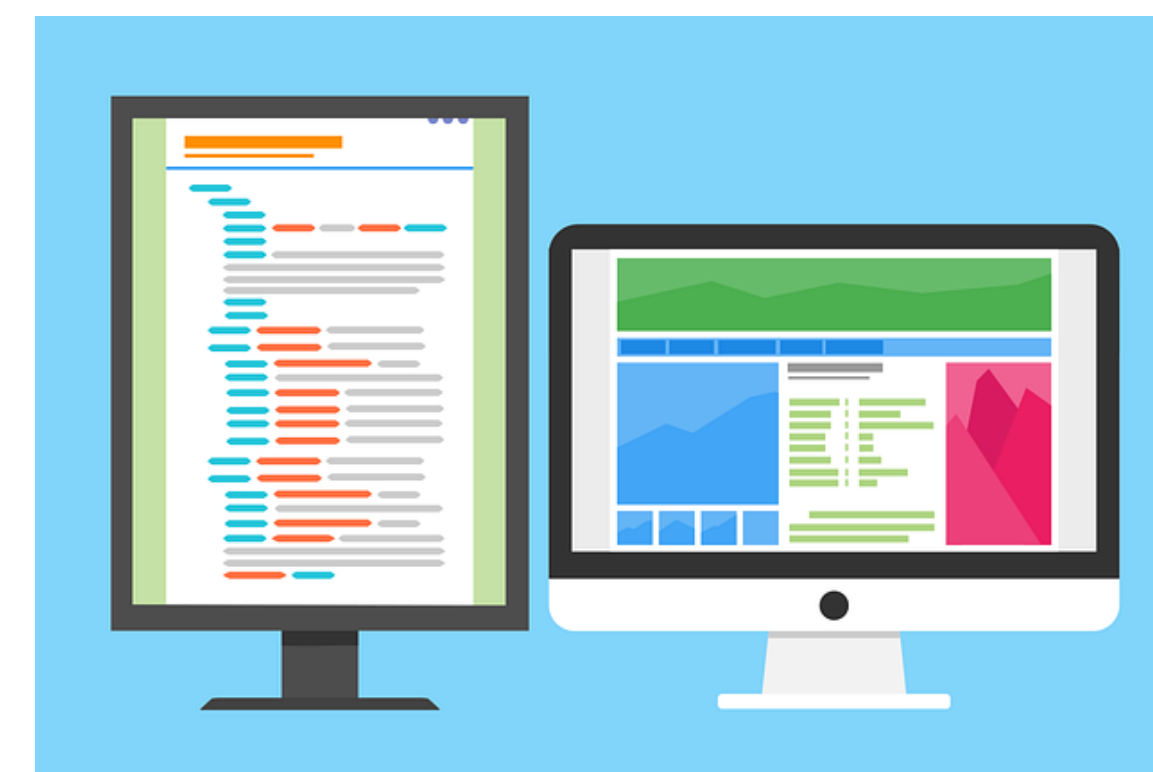
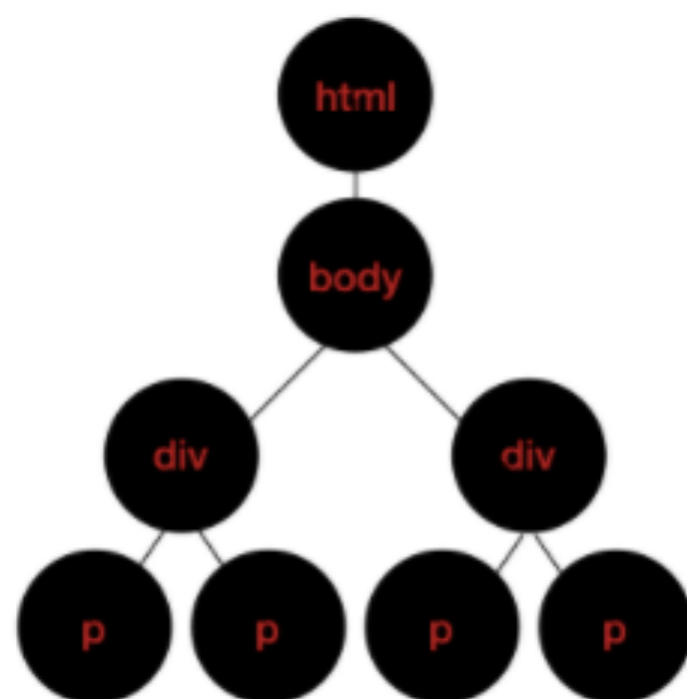
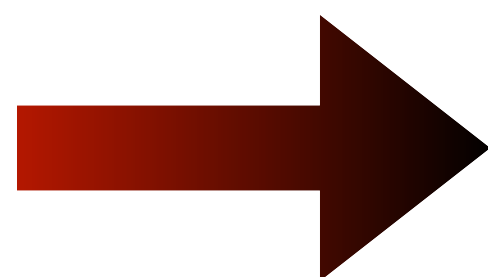


렌더링의 세부과정

브라우저는 HTML 코드를 해석해서 요소들을 트리 형태로 구조화해 표현하는 문서(객체)를 생성한다. 이를 DOM이라 하며, 브라우저는 DOM을 통해 화면에 웹 콘텐츠들을 렌더링한다.

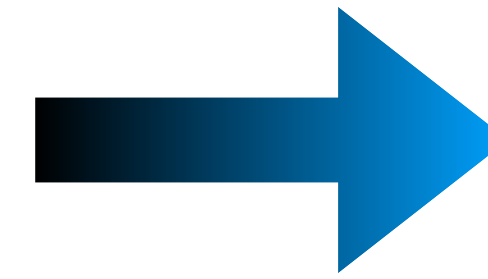
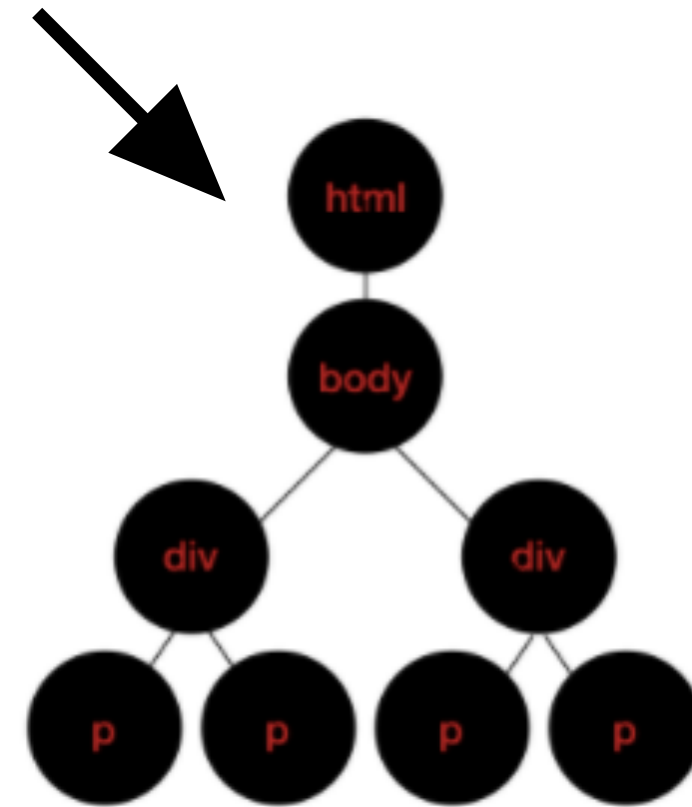
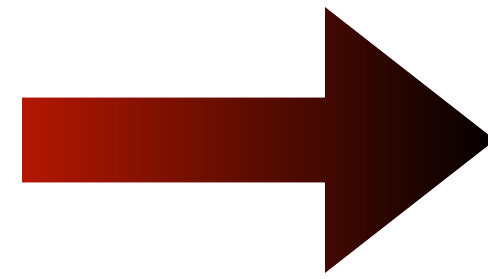
```
<html>
  <body>
    <div>
      <p>헬로우</p>
      <p>헬로우</p>
    </div>
    <div>
      <p>헬로우</p>
      <p>헬로우</p>
    </div>
  </body>
</html>
```







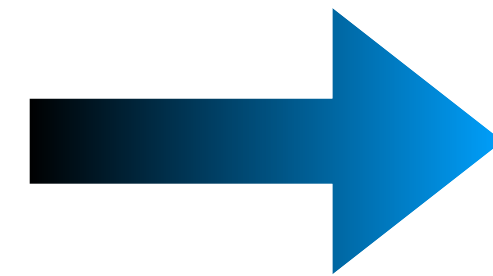
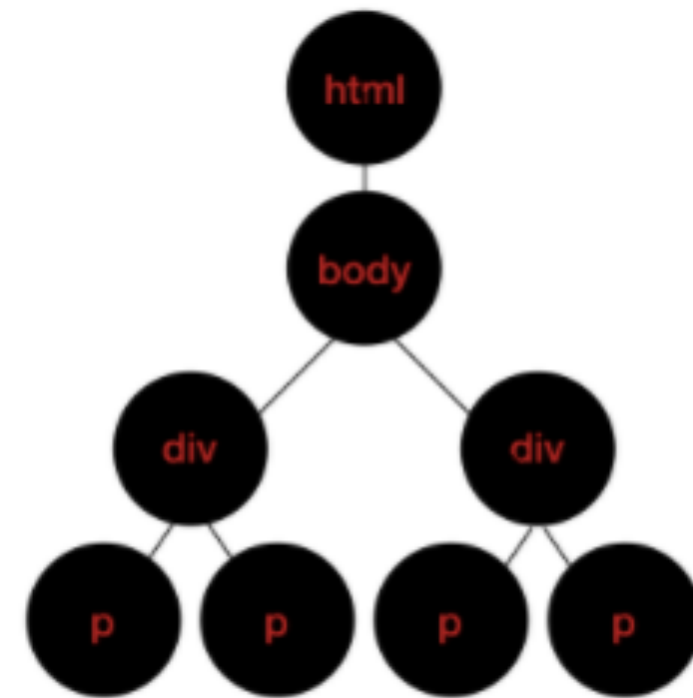
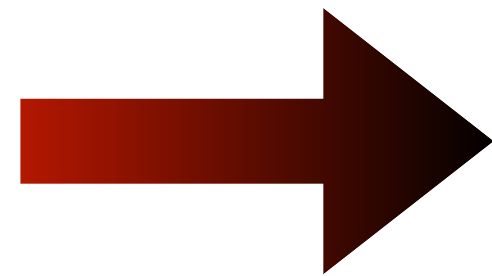
‘DOM’





Document Object Model, DOM

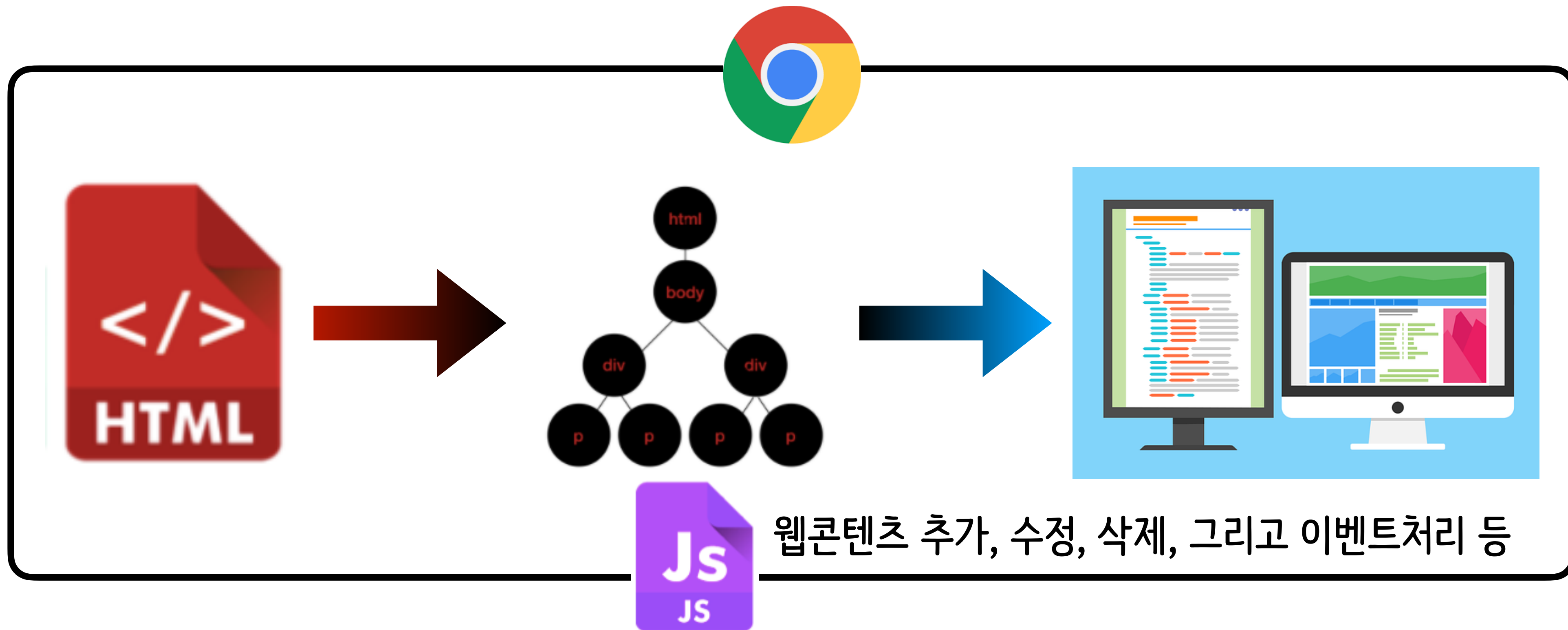
DOM은 웹 콘텐츠를 추가, 수정, 삭제하거나 마우스 클릭, 키보드 타이핑 등 이벤트에 대한 처리를 정의할 수 있도록 제공되는 프로그래밍 인터페이스(interface)이다.





Document Object Model, DOM

DOM은 웹 콘텐츠를 추가, 수정, 삭제하거나 마우스 클릭, 키보드 타이핑 등 이벤트에 대한 처리를 정의할 수 있도록 제공되는 프로그래밍 인터페이스(interface)이다.





DOM에 접근하자

브라우저 객체 window의 document 속성은 창이 포함한 문서를 참조한다.
즉, window.document은 현재 브라우저에 렌더링되고 있는 문서를 의미하며, 이 속성을 이용하면 해당 문서에 접근할 수 있다.

좀 있어 보이게 표현하면?

window.document는 페이지 콘텐츠, 즉 DOM에 대한 진입점 역할을 하는 프로그래밍 인터페이스이다! 이를 이용하면 페이지의 정보를 얻거나 웹 요소를 생성 및 조작할 수 있다.



querySelector & textContent

document의 querySelector 메소드는 선택자를 인자로 전달받아, 전달받은 선택자와 일치하는 문서 내 첫 번째 요소(Element)를 반환한다. 일치하는 요소가 없으면 '없다'라는 의미의 null 데이터를 반환한다.

인자로 전달되는 선택자는 문자열 타입의 '유효한 CSS 선택자'를 의미한다.

```
// p 태그를 선택하자!  
document.querySelector("p");  
  
// id가 text인 요소를 선택하자!  
document.querySelector("#text");  
  
// class가 text인 요소를 선택하자!  
document.querySelector(".text");
```




querySelector & textContent

textContent 속성은 해당 객체(요소)가 포함하고 있는 텍스트 콘텐츠를 표현하는 속성이다. textContent를 통해 요소가 포함한 텍스트를 읽을 수도, 변경할 수도 있다.

```
// p 요소를 반환받아 상수로 이름을 붙인다!  
const p = document.querySelector("p");  
  
// p 요소의 textContent 속성을 콘솔에 출력해보자!  
console.log(p.textContent)  
  
// p 요소의 textContent 값을 변경해보자!  
p.textContent = "텍스트를 이걸로 바꿔!"
```



수고하셨습니다 :)