

혼자 공부하며 함께 만드는

혼공 용어 노트

목차

가나다 순

개발환경 development environment	06	배열 array	11
객체 object	15	변수 variable	10
객체 자료형 object type	15	비동기 프로그래밍	
객체 지향 프로그래밍		asynchronous programming	14
Object-Oriented Programming	20	비파괴적 처리 non destructive	11
구문 오류 syntax error	19	상속 inheritance	20
기본 매개변수 default parameter	13	상수 constant	09
기본 자료형 primitive type	16	생성자 constructor	20
나머지 매개변수 rest parameter	13	새도입 shadowing	14
디자인 패턴 design pattern	22	속성 property	15
라이브러리 library	22	스코프 scope	14
리턴 return	12	식별자 identifier	07
리턴값 return value	13	애플리케이션 application	06
매개변수 parameter	13	엄격모드 strict mode	15
메소드 method	15	에크마스크립트 ECMAScript	06
메소드 체이닝 method chaining	14	예외 exception	19
무한 infinity	16	예외 처리 exception handling	19
문서 객체 document object	17	오버라이드 override	22
문서 객체 모델		유틸리티 라이브러리 utility library	17
DOM: Document Objects Model	17	응용 프로그램 프로그래밍 인터페이스	
문장 statement	07	API: Application Programming Interface	19
반복문 loop statement	11	이벤트 event	18

이벤트 객체 event object	18
이벤트 리스너 event listener	18
이벤트 모델 event model	18
이스케이프 문자 escape character	09
인덱스 index	09
인스턴스 instance	20
일급 객체 first-class object	16
자료형 data type	08
자바스크립트 확장 문법	
JSX: JavaScript XML	23
전개 연산자 spread operator	13
정규 표현식 regular expression	18
정적 속성 static property	21
제이슨 객체	
JSON: JavaScript Object Notation	16
조건문 conditional statement	10
조건부 연산자 conditional operator	11
주석 comment	08
추상화 abstraction	20
컨테이너 container	22
컴포넌트 component	22
코딩 스타일 coding style	07

콘텐츠 전송 네트워크	
CDN: Contents Delivery Network	17
콜백 함수 callback function	14
클래스 class	20
키워드 keyword	07
키 코드 key code	18
템플릿 문자열 template string	09
트리 tree	17
파괴적 처리 destructive	11
표현식 expression	07
프라이빗 속성 private property	21
프레임워크 framework	21
프로토타입 객체 prototype object	16
플럭스 패턴 Flux pattern	23
하이퍼링크 hyperlink	06
함수 function	12
화살표 함수 arrow function	14

ABC 순

abstraction 추상화	20
API: Application Programming Interface	
응용 프로그램 프로그래밍 인터페이스	19
application 애플리케이션	06
array 배열	11
arrow function 화살표 함수	14
asynchronous programming	
비동기 프로그래밍	14
callback function 콜백 함수	14
CDN: Contents Delivery Network	
콘텐츠 전송 네트워크	17
class 클래스	20
coding style 코딩 스타일	07
comment 주석	08
component 컴포넌트	22
conditional operator 조건부 연산자	11
conditional statement 조건문	10
constant 상수	09
constructor 생성자	20
container 컨테이너	22
data type 자료형	08
default parameter 기본 매개변수	14

design pattern 디자인 패턴	22
destructive 파괴적 처리	11
development environment 개발환경	06
document object 문서 객체	17
DOM: Document Objects Model	
문서 객체 모델	17
ECMAScript 에크마스크립트	06
escape character 이스케이프 문자	09
event 이벤트	18
event listener 이벤트 리스너	18
event model 이벤트 모델	18
event object 이벤트 객체	18
exception 예외	19
exception handling 예외 처리	19
expression 표현식	07
first-class object 일급 객체	16
Flux pattern 플럭스 패턴	23
framework 프레임워크	21
function 함수	12
hyperlink 하이퍼링크	06
identifier 식별자	07
index 인덱스	09

Infinity 무한	16	prototype object 프로토타입 객체	16
inheritance 상속	20	regular expression 정규 표현식	18
instance 인스턴스	20	rest parameter 나머지 매개변수	13
JSON: JavaScript Object Notation		return 리턴	12
제이슨 객체	16	return value 리턴값	13
JSX: JavaScript XML 자바스크립트 확장 문법	23	scope 스코프	14
key code 키 코드	18	shadowing 섀도잉	14
keyword 키워드	07	spread operator 전개 연산자	13
library 라이브러리	22	statement 문장	07
loop statement 반복문	11	static property 정적 속성	21
method 메소드	15	strict mode 엄격모드	15
method chaining 메소드 체이닝	14	syntax error 구문 오류	19
non destructive 비파괴적 처리	11	template string 템플릿 문자열	09
object 객체	15	tree 트리	17
object type 객체 자료형	15	utility library 유틸리티 라이브러리	17
Object-Oriented Programming		variable 변수	10
객체 지향 프로그래밍	20		
override 오버라이드	22		
parameter 매개변수	13		
primitive type 기본 자료형	16		
private property 프라이빗 속성	21		
property 속성	15		

01 장

□ 하이퍼링크

hyperlink

[01장 031쪽]

문서 내의 요소에 다른 문서로 이동할 수 있도록 연결해놓은 참조 고리.

□ 애플리케이션

application

[01장 032쪽]

응용 프로그램. 다양한 프로그래밍 언어로 애플리케이션을 만들 수 있다.

ex) 웹, 웹 서버, 모바일, 데스크톱 애플리케이션 등

→ 모바일은 리액트 네이티브, 데스크톱은 일렉트론으로 만들 수 있다.

□ 에크마스크립트

ECMAScript

[01장 036쪽]

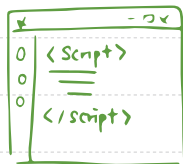
다양한 웹 브라우저에서 자바스크립트 언어가 잘 작동되도록 자바스크립트를 표준화하기 위해 만들어진 자바스크립트 표준 명칭. 대부분의 브라우저에서 ECMAScript 2015(ECMAScript 6 또는 ES6) 이상의 기능이 구현되고 있으며 현재 ECMAScript 2020(ES11)까지 나왔다.

□ 개발환경

development environment

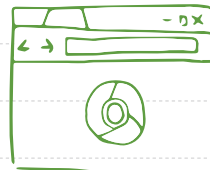
[01장 042쪽]

프로그램을 개발을 할 수 있게 해주는 환경. 코드를 작성하는 텍스트 에디터와 코드 실행기가 필요하다. 웹 개발 시에 텍스트 에디터는 비주얼 스튜디오 코드, 코드 실행기는 구글 크롬 웹 브라우저를 사용할 수 있다.



텍스트 에디터

+



코드 실행기: 웹 브라우저

작성하고 → 실행한다!

□ 코딩 스타일

coding style

[01장 065쪽]

개발자마다 다른 코드 작성 방식을 통일한 코드 작성 규칙, 코딩 컨벤션(coding convention)이라고도 부른다. 언어마다 있는 표준 스타일을 따르거나 팀이나 회사에서 정한 스타일을 따를 수 있다.

ex) 들여쓰기를 2개 할 것인지 4개 할 것인지, 중괄호의 위치는 어떻게 할 것인지, 키워드 뒤에 괄호를 바로 붙일지 공백을 줄지 등...

□ 표현식

expression

[01장 068쪽]

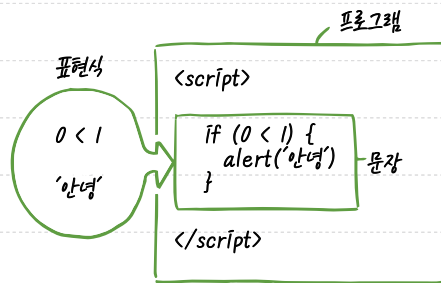
어떠한 값을 만들어내는 간단한 코드. 숫자, 수식, 문자열 등

□ 문장

statement

[01장 069쪽]

표현식이 하나 이상 모인 것. 문장 끝에는 마침표를 찍듯이 세미콜론 또는 줄바꿈을 넣어준다.



□ 키워드

keyword

[01장 070쪽]

특별한 의미가 있는 단어. 언어 내에서 문법적인 용도로 사용되고 있는 단어로 사용자가 지정하는 이름에는 사용할 수 없다.

□ 식별자

identifier

[01장 071쪽]

프로그래밍 언어에서 이름을 붙일 때 사용하는 단어. 주로 변수명이나 함수명 등으로 사용한다.

자바스크립트 식별자를 만들 때 규칙

- 키워드를 사용하면 안된다.

- 숫자로 시작하면 안된다.
- 특수문자는 _와 \$만 허용한다.
- 공백 문자를 포함할 수 없다.

□ 주석

comment

[01장 073쪽]

프로그램 코드를 설명할 때 사용하는 것. 프로그램 진행에는 전혀 영향을 주지 않는다.

그것이 알고싶다

HTML과 자바스크립트의 주석

HTML 태그 주석은 `<!--` 와 `-->` 사이에 글을 입력하여 주석을 사용하고, 자바스크립트 주석은 `//` 뒤에 글을 입력하여 문장 한 줄을 주석으로 쓰거나 `/*` 와 `*/` 사이에 글을 입력하면 `/*` 와 `*/` 사이에 있는 모든 문장이 주석이 된다.

02장

□ 자료형

data type

[02장 084쪽]

프로그램이 처리할 수 있는 자료를 형태에 따라 나누어 놓은 것.

자바스크립트 기본 자료형

- 숫자: 소수점이 있는 숫자와 없는 숫자를 같은 자료형으로 인식한다.
 - 사칙 연산자(+, -, *, /)와 나머지 연산자(%)를 사용한다.
- 문자열: 문자 하나 혹은 문자들의 집합. 큰따옴표 혹은 작은따옴표 안에 문자를 넣어 문자열을 만든다.
 - 문자열 연결 연산자(+), 문자 선택 연산자([])를 사용한다.
- 불: 참과 거짓값을 표현하는 자료형. true와 false를 입력하여 만든다.
 - 비교 연산자(===, !==, >, <, >=, <=), 불 부정 연산자(!), 논리 연산자(&&, ||)를 사용한다.

- undefined: 값이 없는 변수
- null: 빈 값

□ 이스케이프 문자

escape character

[02장 086쪽]

문자열 내부에서 특수한 기능을 수행하는 문자. \ (역슬래시) 기호를 사용한다.

- \": 문자열에서 따옴표 문자 그대로 사용할 수 있다.
- \n: 줄바꿈을 의미한다.
- \t: 탭을 의미한다.
- \\: 역슬래시 문자 그대로 사용할 수 있다.

□ 인덱스

index

[02장 088쪽]

문자열 같은 자료형에서 선택할 문자의 위치를 나타내는 숫자. 자바스크립트는 인덱스를 0부터 센다.

‘안녕하세요’[0] → ‘안’

안	녕	하	세	요
[0]	[1]	[2]	[3]	[4]

↓ 위치를 나타내는 숫자

□ 템플릿 문자열

template string

[02장 098쪽]

문자열 내부에 표현식을 삽입할 때 문자열 연결 연산자(+)를 사용하지 않고 간단하게 문자열과 표현식을 함께 사용할 수 있는 기호.

문자열 내부에 `\${...}` 기호를 사용하여 괄호 안에 표현식을 넣어 사용한다.

↓ 작은따옴표와 혼동하지 말자

□ 상수

constant

[02장 103쪽]

값을 저장하는 식별자.

‘항상 같은 값’으로 이름을 한 번 붙이면 값을 수정할 수 없다.

상수를 만들 때는 const 키워드를 사용한다.

- 선언: 식별자가 어떤 데이터를 가지는지 알려주는 것을 ‘선언한다’라고 한다.

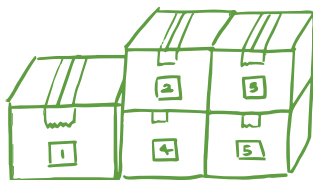
값을 저장하는 식별자.

'변할 수 있는 값'으로 변수를 만들 때는 let 키워드를 사용한다.

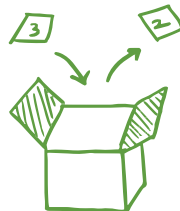
그것이 알고싶다 상수와 변수

상수는 값을 수정할 수 없는 대신 저장 효율이 좋고, 변수는 값을 수정할 수 있는 대신 저장 효율이 나쁘다. '변경할 가능성이 있으면 변수를 사용하고, 그렇지 않으면 상수를 사용한다'라는 기본 규칙을 지켜 사용한다.

const(상수)



let(변수)



03장 ✓

조건에 따라 코드를 실행하거나 실행하지 않도록 할 때 사용하는 구문.

코드가 실행되는 흐름을 변경하는 것을 조건 분기라고 한다.

- if 조건문: 괄호 안의 불 표현식 값이 참이면 중괄호 안의 문장을 실행. 거짓이면 문장을 무시.
- if else 조건문: if 조건문 바로 뒤에 조합하여 사용. if 조건문이 거짓일 때 실행
- 중첩 조건문: 조건문 안에 조건문을 중첩해서 사용하는 것.
- if else if 조건문: 중첩 조건문에서 중괄호를 생략한 형태로 겹치지 않는 3가지 이상의 조건으로 나눌 때 사용.
- switch 조건문: 괄호 안에 비교할 값을 입력하여 특정 값의 조건을 비교할 때 사용한다. break 키워드로 조건문을 끝낸다. default 키워드는 생략할 수 있다.

□ 조건부 연산자

conditional operator

[03장 145쪽]

자바스크립트에서 조건문과 비슷한 역할을 하는 항을 3개 갖는 연산자. 삼항 연산자라고 부르기도 한다.

- 불 표현식 ? 참일 때의 결과 : 거짓일 때의 결과

04장 ✓

□ 배열

array

[04장 160쪽]

여러 개의 변수를 한 번에 선언하여 다룰 수 있는 자료형.

대괄호[]를 사용하여 생성하고 내부의 값을 덱스(,)로 구분한다.

- 요소: 배열 내부에 들어있는 값.

→ 요소, 요소, 요소, ...

□ 비파괴적 처리

non destructive

[04장 169쪽]

처리 후에 원본 내용이 변경되지 않는 처리.

□ 파괴적 처리

destructive

[04장 169쪽]

처리 후에 원본 내용이 변경되는 처리.

□ 반복문

loop statement

[04장 174쪽]

코드를 여러번 혹은 무한히 반복하고 싶을 때 사용하는 구문.

반복문이 무한 반복되는 것을 무한 루프라고 한다.

- for in 반복문: 배열 요소를 하나하나 꺼내서 특정 문장을 실행할 때 사용.
 - 반복 변수에는 인덱스로 접근한다.
- for of 반복문: 반복 변수에 인덱스 대신에 요소의 값을 기준으로 반복할 때 사용.

- for 반복문: 특정 횟수만큼 반복하고 싶을 때 사용하는 범용적인 반복문.
 - 변수는 let 키워드로 선언한다.
- while 반복문: 불 표현식이 true면 계속해서 문장을 실행. 반복 횟수를 모르거나 무한 루프를 만들 때 사용한다.
- break 키워드: switch 조건문이나 무한 루프 반복문을 벗어날 때 사용한다.
- continue 키워드: 현재 반복 작업을 멈추고 반복문의 처음으로 돌아가 다음 반복 작업을 진행한다.

05장

□ 함수

function

[05장 197쪽]

코드의 집합을 나타내는 자료형으로 '기능'을 의미한다.

- 익명 함수: 이름이 붙어있지 않은 함수. `function () {}` 형태로 만든다.
 - 이름이 없기 때문에 변수에 넣어서 사용한다.
- 선언적 함수: 이름이 붙어있는 함수. `function 함수 이름 () {}` 형태로 만든다.
 - function 키워드를 이용하여 함수를 선언한다.

그것이 알고싶다 **익명 함수와 선언적 함수**

익명 함수와 선언적 함수의 가장 큰 차이는 호출 순서의 차이다. 익명 함수는 코드를 위에서 아래로 읽은 순서대로 실행하기 때문에 함수를 호출하기 전에 먼저 선언해야 실행할 수 있다. 하지만 선언적 함수는 웹 브라우저에서 코드를 위에서 아래로 읽기 전에 선언적 함수부터 읽기 때문에 함수를 선언하기 전에 호출해도 오류없이 실행된다.

□ 리턴

return

[05장 200쪽]

함수를 실행했던 위치로 돌아가게 하는 것.

리턴값을 가지는 함수는 반드시 리턴할 때 반환하는 값이 있어야 한다.

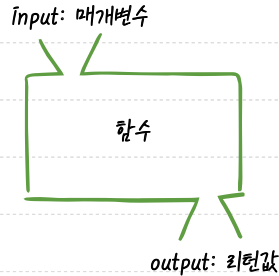
- 조기 리턴: 함수 내 필요한 위치에서 return 키워드를 사용하는 것.

□ 리턴값

return value

[05장 200쪽]

함수의 실행 결과값. 함수의 최종 결과.

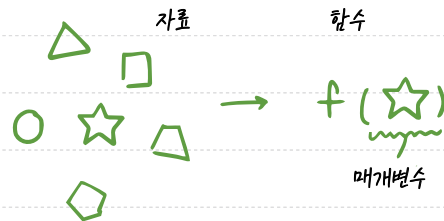


□ 매개변수

parameter

[05장 200쪽]

함수를 호출할 때 필요한 데이터를 외부에서 받기 위해 사용하는 것. 괄호 안에 적는다.



□ 나머지 매개변수

rest parameter

[05장 206쪽]

호출할 때 매개변수의 개수가 고정적이지 않은 함수를 가변 매개변수 함수라고 부르는데 자바스크립트에서는 나머지 매개변수로 구현한다. 나머지 매개변수는 매개변수 앞에 ...을 붙여서 선언한다.

- arguments: 구 버전 자바스크립트에서

가변 매개변수 함수를 구현할 때 사용한 변수.

↓
나머지 매개변수는 하나만!
매개변수 마지막 위치에서만 OK!

□ 전개 연산자

spread operator

[05장 210쪽]

배열을 함수의 매개변수로써 전개하고 싶을 때 사용한다. 배열 앞에 ...을 붙여서 사용한다.

→ 배열 요소 하나하나가 전개된다. [1, 2, 3, 4]

□ 기본 매개변수	default parameter	[05장 212쪽]
	매개변수에 기본값을 지정할 때 사용한다. 기본 매개변수는 매개변수의 오른쪽에 사용하여 구현한다. 함수 이름(매개변수, 매개변수=기본값)	
	구 버전 자바스크립트에서는 짧은 조건문을 사용해서 기본 매개변수를 구현했다.	
□ 비동기 프로그래밍	asynchronous programming	[05장 220쪽]
	요청과 결과가 동시에 일어나지 않는 것을 의미한다. 코드가 작성된 순서대로 실행되는 것이 아니라 행위를 요청한 뒤, 그 결과를 기다리지 않고 다른 코드를 실행하여 실행 흐름을 방해하지 않고 그대로 흐름을 이어가는 것을 말한다.	
□ 콜백 함수	callback function	[05장 221쪽]
	함수를 매개변수로 전달하는 함수를 말한다.	
□ 화살표 함수	arrow function	[05장 225쪽]
	콜백 함수를 쉽게 입력하는 함수 생성 방법. () => {} 형태로 함수를 만들고, 리턴값만을 가지는 함수라면 (매개변수) => 리턴값으로 간편하게 사용할 수 있다.	
□ 메소드 체이닝	method chaining	[05장 226쪽]
	어떤 메소드가 리턴하는 값을 기반으로 여러 메소드를 이어서 호출하는 문법. 코드가 간결해서 하나의 문장처럼 읽히게 하는 장점이 있다.	
□ 스코프	scope	[05장 230쪽]
	변수에 접근할 수 있는 범위. 중괄호나 함수를 사용해서 블록을 만들어 준다. <ul style="list-style-type: none"> 전역 스코프: 어느 곳에서도 해당 변수에 접근 할 수 있다. 지역 스코프: 범위를 벗어난 곳에서는 해당 변수에 접근할 수 없다. 	
□ 섀도잉	shadowing	[05장 231쪽]
	함수 외부에서 선언한 변수를 함수 내부에서도 같은 이름으로 변수를 선언하여 사	

용할 때, 함수 내부의 변수가 함수 외부의 변수를 가리는 현상. 해당 함수 내부에서 선언된 변수만 인식하기 때문에 변수가 서로 충돌하지 않는다.

□ 엄격모드

strict mode

[05장 233쪽]

코드를 엄격하게 검사하는 기능. 기존의 자바스크립트 문법을 엄격하게 적용하여 평소에 무시하던 오류를 찾아내기 때문에 보다 안전한 코드를 작성할 수 있다. 스크립트 시작 혹은 함수 시작 부분에 'use strict'라는 문자열을 선언하여 사용한다.

06장

□ 객체

object

[06장 244쪽]

여러 가지 속성을 가질 수 있는 대상.

□ 속성

property

[06장 248쪽]

객체 내부에 있는 값.

배열 내부에 있으면 요소, 객체 내부에 있으면 속성이라고 한다.

- delete 키워드: 객체의 속성을 제거할 때 사용.

□ 메소드

method

[06장 248쪽]

객체의 속성 중에 함수 자료형인 속성.

- this 키워드: 객체 내부의 메소드에서 자기 자신이 가진 속성을 표시할 때 사용.

□ 객체 자료형

object type

[06장 256쪽]

객체를 기반으로 하는 자료형. new 키워드를 사용하여 생성한다.

□ 일급 객체

first-class object

[06장 258쪽]

자바스크립트에서 함수는 객체의 특성을 모두 갖고 있으므로 일급 객체에 속한다.
다른 객체들에게 적용가능한 연산을 모두 지원하는 객체

□ 기본 자료형

primitive type

[06장 252쪽]

실체가 있는 것(undefined와 null이 아닌 것) 중에 객체가 아닌 것.

객체가 아니기 때문에 속성과 메소드를 추가할 수 없다. 숫자, 문자열, 불

- 승급: 기본 자료형의 속성이나 메소드를 사용하면 일시적으로 객체 자료형으로 변화하는 것.

ex) 숫자 자료형에 속성이나 메소드를 사용하면 일시적으로 Number 객체로 변환되면서, 기본 자료형이지만 속성이나 메소드를 사용할 수 있다. 하지만 직접 추가한 것이므로 추가하자마자 버려진다.

이때 프로토타입으로 바꾸면 속성과 메소드를 추가할 수 있다.

□ 프로토타입 객체

prototype object

[06장 262쪽]

객체의 틀. 프로토타입 객체에 속성과 메소드를 추가하면 모든 객체에서 해당 속성과 메소드를 사용할 수 있다. → 기본 자료형의 객체화!

□ 무한

Infinity

[06장 267쪽]

무한대 수를 나타내는 값.

□ 제이슨 객체

JSON: JavaScript Object Notation

[06장 272쪽]

자바스크립트의 배열과 객체를 활용해 자료를 표현하는 방식. 이름과 값이 쌍으로 이루어진 데이터. 객체를 문자열로 변환할 때 JSON.stringify() 메소드를 사용한다.

- 값을 표현할 때 문자열 숫자, 불 자료형만 사용 가능하다. (함수 불가)
- 문자열은 반드시 큰따옴표(" ")로 만든다.
- 키(key)에도 반드시 따옴표를 붙인다.

□유틸리티	utility library	[06장 280쪽]
라이브러리	개발할 때 보조적으로 사용하는 함수를 제공해주는 라이브러리. underscore, Lodash 등의 다양한 라이브러리가 있다.	

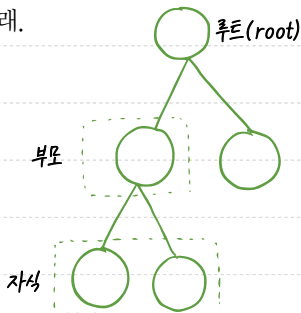
□콘텐츠 전송	CDN: Contents Delivery Network	[06장 282쪽]
네트워크	콘텐츠 전송 네트워크. 전 세계 여러 지역에 전송할 콘텐츠를 창고처럼 준비해두고, 멀리 떨어져 있는 사용자가 콘텐츠를 요청했을 때 가장 가까운 지역에서 콘텐츠를 빠르게 제공하는 기술.	

07장

□문서 객체	document object	[07장 306쪽]
	HTML 페이지에 있는 html, body, title, h1, div, span 등의 요소를 자바스크립트에서는 문서 객체라고 말한다.	

□문서 객체 모델	Document Objects Model	[07장 306쪽]
	문서 객체를 조합해서 만든 전체적인 형태. 약자로 'DOM'이라고 한다.	

□트리	tree	[07장 319쪽]
	<p>비선형 자료구조의 일종. 나무처럼 생긴 것에서 유래.</p> <ul style="list-style-type: none"> • 부모: 어떤 문서 객체의 위에 있는 것. • 자식: 어떤 문서 객체의 아래에 있는 것. 	



□ 이벤트	event	[07장 323쪽]
	<p>웹 브라우저에서 발생하는 사용자의 동작. 마우스 클릭, 스크롤, 키보드 입력, 페이지 전환 등</p> <ul style="list-style-type: none"> • DOMContentLoaded 이벤트: HTML 페이지의 모든 문서 객체(요소)를 웹 브라우저가 읽었을 때 발생시키는 이벤트. 	
□ 이벤트 리스너	event listener	[07장 323쪽]
	이벤트가 발생할 때 실행하는 함수. 이벤트 리스너 또는 이벤트 핸들러라고 한다.	
□ 이벤트 모델	event model	[07장 329쪽]
	<p>이벤트를 연결하는 방법.</p> <ul style="list-style-type: none"> • 고전 이벤트 모델: 문서 객체가 갖고 있는 onOO으로 시작하는 속성에 함수를 할당하여 이벤트를 연결하는 방법. • 인라인 이벤트 모델: onOO으로 시작하는 속성을 HTML 요소에 직접 넣어 이벤트를 연결하는 방법. 	
□ 이벤트 객체	event object	[07장 329쪽]
	이벤트 리스너의 첫 번째 매개변수로 이벤트와 관련된 정보가 들어있다.	
□ 키 코드	key code	[07장 332쪽]
	키보드 이벤트가 발생할 때 어떤 키를 눌렀는지 키 값을 구분할 수 있는 코드.	
□ 정규 표현식	regular expression	[07장 342쪽]
	<p>특정한 규칙을 가지는 문자열의 집합.</p> <p>유효성 검사를 쉽고 확실하게 구현할 수 있다.</p> <p>ex) 이메일, 비밀번호 검사 등</p>	

애플리케이션 소프트웨어에서 프로그램들이 서로 상호작용하도록 도와주는 매개체.

08장

프로그램 실행 중에 발생하는 오류. 예외 또는 런타임 오류라고 부른다.

문법적인 오류를 제외하고 실행 중간에 발생하는 오류를 '예외'라고 부르는데, 예외가 발생하여 프로그램이 중단되지 않도록 처리하는 것을 말한다.

- 조건문 사용: 예외가 발생하지 않게 만드는 기본 예외 처리.
- try catch finally 구문: try 구문 안에서 예외가 발생하면 catch 구문에서 처리하는 고급 예외 처리.
 - finally 구문은 예외 발생과 상관 없이 무조건 실행되기 때문에 필요한 경우에만 사용한다.
- throw 키워드: 예외를 강제로 발생시키는 키워드. 예외가 발생하면서 프로그램이 중단된다.

프로그램 실행 전에 발생하는 오류.

구문 오류가 있으면 프로그램 자체가 실행되지 않는다.

09장

□ 클래스 **class** [09장 386쪽]

객체를 효율적이고 안전하게 만들기 위해 만들어진 문법. 객체에 포함할 변수와 함수가 미리 정의되어 있어 객체의 설계도에 해당한다.

□ 추상화 **abstraction** [09장 387쪽]

프로그램에서 필요한 요소만을 사용해서 객체를 표현하는 것.

□ 객체 지향 프로그래밍 **Object-Oriented Programming** [09장 394쪽]

객체를 정의하고, 객체를 활용해서 프로그램을 만드는 것. C++, C#, 자바, 루비, 코틀린, 스윕트, PHP 등은 모두 클래스 기반의 객체 지향 프로그래밍 언어이다.

□ 인스턴스 **instance** [09장 394쪽]

클래스를 기반으로 만든 객체.

그것이 알고싶다 **클래스, 객체, 인스턴스**

객체는 실제로 존재하는 것이다. 속성을 가진 객체는 클래스를 이용하여 인스턴스로 구현할 수 있다.

□ 생성자 **constructor** [09장 396쪽]

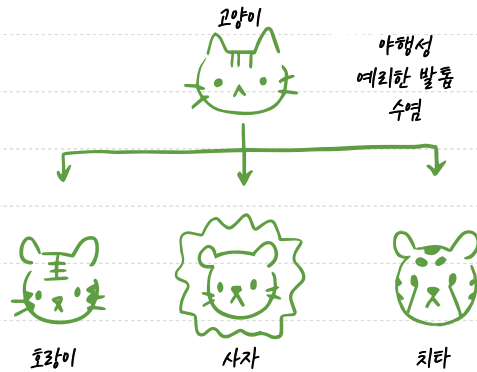
객체가 생성될 때 호출되는 함수. 클래스 이름으로 호출한다. 클래스를 기반으로 인스턴스를 생성할 때 처음 호출되는 메소드이기 때문에 생성자에서는 속성을 추가하는 등 객체의 초기화 처리를 한다.

□ 상속 **inheritance** [09장 403쪽]

클래스의 선언 코드를 중복해서 작성하지 않게 코드의 생산 효율을 올리는 문법.

어떤 클래스가 가지고 있는 유산(속성과 메소드)을 다른 클래스에게 물려주는 형태.

- 부모 클래스: 유산을 주는 클래스
- 자식 클래스: 유산을 받는 클래스



□ 프레임워크

framework

[09장 407쪽]

프로그램을 개발할 때 사용하는 거대한 규모의 클래스, 함수, 도구의 집합

□ 프라이빗 속성

private property

[08장 407쪽]

클래스 사용자가 클래스의 속성(또는 메소드)을 의도하지 않은 방향으로 사용하는 것을 막아 클래스의 안정성을 확보한다. 프라이빗 메소드라고도 한다. 속성이나 메소드 이름 앞에 #을 붙여 사용한다.

그것이 알고싶다 게터와 세터

프라이빗 속성을 사용하면 외부에서는 속성에 아예 접근할 수 없는 문제가 발생한다. 이에 따라 속성을 읽고 쓸 수 있는 메소드를 제공한다.

- 게터(getter): 속성 값을 확인할 때 사용하는 메소드. get이름()
- 세터(setter): 속성에 값을 지정할 때 사용하는 메소드. set이름()

□ 정적 속성

static property

[09장 416쪽]

인스턴스를 만들지 않고 사용할 수 있는 속성. 정적 메소드라고도 한다. 일반적인 변수와 함수처럼 사용할 수 있으며 클래스 이름 뒤에 점을 찍고 속성과 메소드를 사용한다.

인스턴스에서는 호출할 수 없다.

□ 디자인 패턴

design pattern

[09장 416쪽]

효율적으로 프레임워크를 개발할 수 있게 고안한 패턴. 개발하는 과정에서 빈번하게 발생하는 문제들이 발생했을 때 해결책을 재사용할 수 있도록 만든 패턴이다.

□ 오버라이드

override

[09장 419쪽]

부모가 갖고 있는 함수를 자식에서 다시 선언해서 덮어쓰는 것.

10장

□ 라이브러리

library

[10장 428쪽]

프로그램 개발에 쓰이는 하부 프로그램.

- 리액트 라이브러리: 사용자 인터페이스(UI)를 쉽게 구성할 수 있도록 도와주어 대규모 프론트엔드 웹 애플리케이션을 체계적으로 개발할 수 있다.

□ 컴포넌트

component

[10장 430쪽]

리액트에서 화면에 출력되는 요소를 말한다.

- 루트 컴포넌트: 가장 최상위에 배치하는 컴포넌트
- 클래스 컴포넌트: 사용자가 직접 클래스로 만드는 컴포넌트
- 함수 컴포넌트: 사용자가 직접 함수로 만드는 컴포넌트

□ 컨테이너

container

[10장 430쪽]

컴포넌트를 출력할 상자.

 컴포넌트를 출력할 때는 `ReactDOM.render()` 메소드 사용.

확장 문법

리액트를 위해서 개발된 자바스크립트 확장 문법. 자바스크립트 코드 내부에 HTML 태그 형태로 컴포넌트를 만들 수 있다. JSX 자바스크립트 확장 문법이라고 한다. 단순히 태그를 만드는 기능 이외에도 태그 내부에 표현식을 삽입해서 출력하는 기능도 제공한다.

- 바벨(babel): JSX 코드를 읽고 일반적인 자바스크립트 문법으로 변환해주는 라이브러리. 자바스크립트 코드 내부에 HTML 코드를 사용할 수 있다.

```
<script type="text/babel"></script>
```

디자인 패턴의 일종. Flux 아키텍처라고도 한다.

- MobX 라이브러리: Flux 패턴으로 데이터를 쉽게 관리할 수 있다.

MEMO

Handwriting practice area with horizontal dashed lines.

혼자 공부하는 사람들을 위한 용어 노트

[illegible]