**JavaScript 함수**

**<<01. 함수 복습>>**

**\* 값이 필요할 때마다 함수 호출 vs 변수에다 함수 결과값 넣어서 사용:**

- 함수 호출할 때마다 자원이 사용되므로, 변수에다 값을 넣어 변수만 사용하는 것이 좋음

- 반복적: 변수에다 넣어서 사용

- 일회성: 함수 호출

**\* 함수 선언:** 일반적인 방법

**\* 함수 표현**

- function 뒤에 함수명 있으면: 기명 함수

- function 뒤에 함수명 없으면: 익명 함수

**\* return:** 해당 부분에서 종료됨. 그 이후 코드는 실행 안 됨.

**\* arguments:** 매개변수를 지정하지 않아도 함수 내에서 사용할 수 있게 만들어진 객체

**<<02. 화살표 함수>>**

**\* 화살표 함수**

**-** => === function

- 축약형으로 표현 가능

- 매개변수가 하나일 때는 ()도 생략 가능

- 축약형으로 표현할 때) 객체 표현을 {}로 하기 때문에 블록의 {}을 객체 표현의 {}로 착각할 수 있음 -> 그렇기 때문에 소괄호()로 묶은 다음에 블록의 {}을 표현해줘야 함

**<<03. IIFE>>**

**\* 즉시실행함수(IIFE, Immediately-Invoked Function Expression):**

- 소괄호를 열고 이름 없는 즉시실행함수를 작성, 그 후 소괄호 ()를 붙여 실행(혹은 함수 닫는 소괄호 안에 작성해도 됨)

- 두번째 방법을 권장

**<<04. 호이스팅>>**

**\* 호이스팅(Hoisting):** 함수 선언부가 유효범위 최상단으로 끌어올려지는 현상

- 함수 표현에서는 사용 불가

- 함수 선언에서는 사용 가능

\* 호이스팅이 유용할 때:

- 함수 호출이 먼저 되고 함수 선언이 밑에 있으면 코드 해석함에 있어 편함

- (어디서 실행될지도 모르는 함수를 먼저 선언하니…)

**<<05. 타이머 함수>>**

**\* 타이머 함수**

**- setTimeout(함수, 시간):** 일정 시간 후 함수 실행

**- setInterval(함수, 시간):** 시간 간격마다 함수 실행

**- clearTimeout():** 설정된 Timeout 함수를 종료

**- clearInterval():** 설정된 Interval 함수를 종료

**\* setTimeout, clearTimeout():**

**\* setInterval, clearInterval():**

**<<06. 콜백>>**

**\* 콜백(Callback):** 함수의 인수로 사용되는 함수

- 호출되지 않은 함수를 매개변수로 받아 원하는 처리 이후 인수로 원하는 장소에 넣어서 사용  
- 특정 실행 위치를 보장하는 용도로 활용됨