#5.0

Interval: 매번 일어나야 하는 무언가

setInterval( ) : 두 개의 argument를 받음

- 지정한 시간마다 반영 🡪 1초 간격으로 설정할 경우 실시간으로 반영되는 것 같은 효과 제공!!

- 첫 번째 argument: 내가 실행하고자 하는 function

- 두 번째 argument: 호출되는 function 간격을 몇 ms로 할 것인지 작성 (5000ms=5초)

#5.1

setTimeout( ): 두 개의 argument를 받음

- 첫 번째 argument: 내가 호출하고자 하는 function

- 두 번째 argument: 호출되는 function 간격을 몇 ms로 할 것인지 작성 (5000ms=5초)

#5.2

padStart(a, “b”)

- 길이가 a로 되어야하고 공란이 생길경우 b로 채울 것

- string을 보다 길게 만들어야 할 때 사용

\*date.getHours( ) 에는 padStart( )를 사용할 수 x: string과 number로 형식이 다르기 때문

#6.0

Math.random( ): 0부터 1 사이의 랜덤한 숫자 제공

* Math.random( ) \* 10 : 0부터 10 사이의 랜덤한 숫자 제공

Math.ceil( ) : 숫자를 올림해서 반환

Math.floor( ): 숫자를 내림해서 반환

Math.round( ): 숫자를 반올림해서 반환

Array.length: array의 개수를 알려줌

#6.1

랜덤하게 불러와야하기 때문에 HTML이 아닌 Javascript로 img 파일 src 생성

마지막에 잊지말고 생성한 변수를 body에 append 해야함 (prepend도 사용 가능)

#7.0

function handleToDoSubmit(event) {

    event.preventDefault();

    const newTodo = toDoInput.value;

    toDoInput.value = "";

}

toDoInput.value 를 비웠다고 해서 newTodo가 비워지지 않는다!

#7.1

function paintToDo(newTodo) {

    const li = document.createElement("li");

    const span = document.createElement("span");

    li.appendChild(span);

    span.innerText = newTodo;

    console.log(li);

}

Span을 만들고 그 span을 li 내부에 집어넣음

#7.2

Button은 click event 와 짝궁

Button이 기능을 하기 위해서는 eventListener 사용

모든 button이 같은 기능 🡪 어떤 li에 적용해야 할지 알 수 x

ㄴpath에서 확인 가능

parentNode: 클릭된 button의 부모 확인

parentElement: 클릭된 element의 부모 확인

#7.3

JSON.stringify: 어떤 java간에 string으로 만들어줌

#7.4

JSON.parse: 실제로 무언가를 할 수 있는 배열로 변경됨

#7.5

-array function

-sayHello function

-> 위 두 개는 같음! Array funciton이 상대적으로 더 짧다

#7.6

function deleteToDo(event) {

    const li = event.target.parentElement;

    console.log(li.id);

    li.remove();

}

Li의 id를 얻을 수 있는 기능

#7.7

\*\*paintToDo 정확하게 이해할 것

if(savedToDos !== null) {

    const parsedToDos = JSON.parse(savedToDos);

    toDos = parsedToDos;

    parsedToDos.forEach(paintToDo);

}

forEach : paintToDo를 parsedToDos 배열의 요소마다 실행

ㄴpaintToDo를 기본적으로 실행함

#7.8

function deleteToDo(event) {

    const li = event.target.parentElement;

    li.remove();

    toDos = toDos.filter((toDo) => toDo.id !== parseInt(li.id));

}

클릭한 li.id와 다른 toDo는 남김