노드 내장 객체 알아보기 필기노트

2021년 1월 15일 금요일 오후 10:08

1. 내장 객체 알아보기 전 주의사항
   외울 필요 X, 이런게 있구나만 생각하자, 필요할 경우 다시 확인하자(사전 느낌)

   a. global
      노드의 전역 객체로, 브라우저의 window 같은 역할
      모든 파일에 접근 가능, window처럼 생략도 가능
      참고로 node에서는 document같은 것은 동작하지 않음
      대표적 예시
   - global.console.log -> console.log
   - global.require -> require
   - global.module.exports -> module.exports

   b. console 객체
      브라우저의 console 객체와 매우 유사
      console.time, console.timeEnd : 시간 로깅
      console.error : 에러 로깅
      console.log : 평범한 로그
      console.dir : 객체 로깅
      console.trace : 호출 스택 로깅

소스 코드	실행 결과

```
const string = 'abc';
const number = 1;
const boolean = true;
const obj = {
    outside: {
        inside: {
            key: "value",
        },
    },
};
console.time("전체 시간");
console.log("평범한 로그입니다. 쉼표로 구분해 여러 값을 찍을 수 있습니다.");
console.log(string, number, boolean);
console.error("에러 메세지는 console.error에 담아주세요");
console.table([
    {
        name: '제로',
        birth: 1994
    },
    {
        name:'hero',
        birth: 1988
    }
]);
console.dir(obj, {colors:false, depth:2});
console.dir(obj, {colors:true, depth:1});
console.time('시간 측정');
for(let i=0; i<100000; i++){}
console.timeEnd('시간 측정');
function b(){
    console.trace('에러 위치 추적');
}
function a(){
    b();
}
a();
console.timeEnd('전체 시간');
```

```
평범한 로그입니다. 쉼표로 구분해 여러 값을 찍을 수 있습니다.
abc 1 true
에러 메세지는 console.error에 담아주세요
┌─────────┬──────┬─────────┐
| (index) | name | birth |
├─────────┼──────┼─────────┤
|   0   | '제로' | 1994  |
|   1   | 'hero' | 1988  |
└─────────┴──────┴─────────┘
{ outside: { inside: { key: 'value' } } }
{ outside: { inside: [Object] } }
시간 측정: 3.925ms
Trace: 에러 위치 추적
    at b (e:\프로그래밍\Nodejs 교과서\3. 노드 기능\global과 콘솔,타이머\console.js:35:13)
    at a (e:\프로그래밍\Nodejs 교과서\3. 노드 기능\global과 콘솔,타이머\console.js:38:5)
    at Object.<anonymous> (e:\프로그래밍\Nodejs 교과서\3. 노드 기능\global과 콘솔,타이머
\console.js:40:1)
    at Module._compile (internal/modules/cjs/loader.js:1137:30)
    at Object.Module._extensions..js (internal/modules/cjs/loader.js:1157:10)
    at Module.load (internal/modules/cjs/loader.js:985:32)
    at Function.Module._load (internal/modules/cjs/loader.js:878:14)
    at Function.executeUserEntryPoint [as runMain] (internal/modules/run_main.js:71:12)
    at internal/main/run_main_module.js:17:47
전체 시간: 107.192ms
```

   c. 타이머 메서드
      set메서드에 clear메서드가 대응됨
      set메서드의 리턴값(아이디)를 clear메서드에 넣어 취소
      setTimeOut(콜백함수, 밀리초) : 주어진 밀리초(1/1000초)이후에 콜백 함수 실행
      setInterval(콜백함수, 밀리초) : 주어진 밀리초(1/1000초)마다 콜백 함수 반복하여 실행
      setImmediate(콜백 함수) : 콜백 함수를 즉시 실행
      clearTimeout(아이디) : setTimeout을 취소
      clearInterval(아이디) : setInterval을 취소
      slearImmediate(아이디) : setImmediate를 취소

setTImeout(콜백, 0)보다 setImmediate(콜백)을 권장함

소스 코드	실행 결과

const timeout = setTimeout(()=>{
  console.log('1.5초 후 실행');
},1500);
const interval = setInterval(()=>{
  console.log('1초마다 실행');
},1000);
const timeout2 = setTimeout(()=>{
  console.log('실행되지 않습니다');
},3000);
setTimeout(()=>{
  clearTimeout(timeout2);
  clearInterval(interval);
},2500);
const immediate = setImmediate(()=>{
  console.log('즉시 실행');
});
const immediate2 = setImmediate(()=>{
  console.log('실행하지 않습니다');
})
clearImmediate(immediate2);
	즉시 실행
	1초마다 실행
	1.5초 후 실행
	1초마다 실행