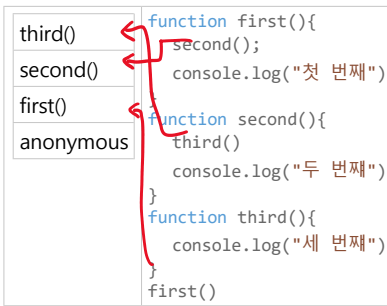


필기 노트

2021년 1월 12일 화요일 오후 7:35

1. 호출 스택

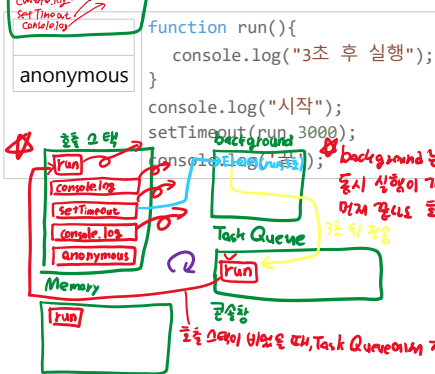


anonymous는 가상의 전역 컨텍스트

자바스크립트는 함수를 스택(LIFO)방식으로 호출 후 실행

Last In First Out

2. 이벤트 루프

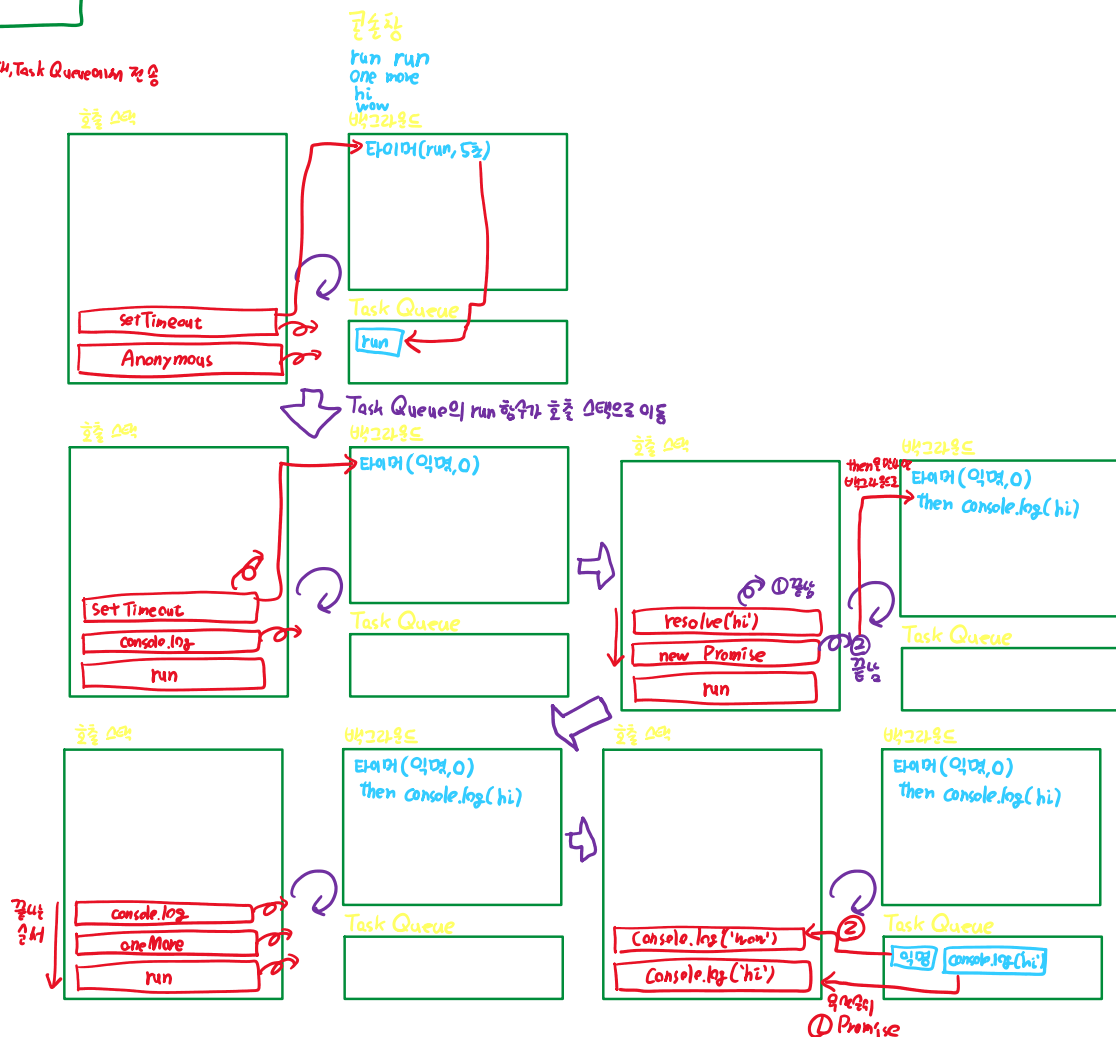


- 복잡한 예제 분석

```

function oneMore(){
  console.log('one more');
}
function run(){
  console.log('run run');
  setTimeout(() => {
    console.log('wow')
  }, 0);
  new Promise((resolve) => {
    resolve('hi');
  })
  .then(console.log);
  oneMore();
}
  
```

promise then/catch, setTimeout(run, 0), process.nextTick



3. ES2015+ 문법

- var, let, const

var : 문법에서 사라져도 지장 X, 레거시 코드 분석 시에는 특성을 알아야 함

var은 함수 스코프, let 및 const는 블록 스코프

var은 블록 스코프를 무시하므로 블록 밖에서도 접근 가능

let, const는 블록 스코프이므로 블록 밖에서 접근 불가

d	var	let, const
if, while, for	무시	접근 불가
function	접근 불가	접근 불가

const는 변수 자체에 =을 한번만 붙일 수 있다

const b = {name:'zerocho'};

b.name = 'hello; 는 가능

- 템플릿 문자열

var result = '이 과자는' + won + '원 입니다';

-> const result = `이 과자는 \${won}원 입니다.`;

함수 호출 시에도 a(); 대신 a`도 가능함

- 객체 리터럴

옛 방식	새 방식
<pre>var sayNode = function(){ console.log('Node'); }; var es = 'ES'; var oldObject = { sayJS : function(){ console.log('JS'); }, sayNode : sayNode, }; oldObject[es + 6] = 'Fantastic'; oldObject.sayNode(); //Node oldObject.sayJS(); //JS console.log(oldObject.ES6);</pre>	<pre>let sayNode = function(){ console.log('Node'); }; const newObject = { sayJS(){ console.log('JS'); }, sayNode, [es + 6] : 'Fantastic', }; newObject.sayNode(); //Node newObject.sayJS``; // JS console.log(newObject.ES6);</pre>

- 화살표 함수

```
function add1(x, y){
  return x + y;
}
```

// add1을 화살표 함수로 나타냄

```
const add2 = (x,y) => {
  return x + y;
}
```

// return만 있는 경우 생략 가능

```
const add3 = (x,y) => x + y;
```

// return이 생략된 경우 본문을 소괄호로 감싸줄 수 있음

```
const add4 = (x,y) => (x + y);
```

// 아래 두 함수는 동일한 기능

```
function not1(x){
  return !x;
}
```

```
const not2 = x => !x;
```

생각이 많으면
해결할 수 있음

let relationship1 = {	const relationship2 = {
<pre>name: 'zero', friends: ['hero', 'hero', 'xero'], logFriends: function(){ var that = this; this.friends.forEach(function(friend){ console.log(that.name, friend); }) }</pre>	<pre>name: 'zero', friends: ['hero', 'hero', 'xero'], logFriends(){ this.friends.forEach(friend => { console.log(this.name, friend); }) }</pre>
relationship1.logFriends();	relationship2.logFriends();

예시

```
this
button.addEventListener('click',function(){
  console.log(this.textContent);
})
btn2.addEventListener('click',()=>{
  console.log(this.textContent);
})
```

```
btn2.addEventListener('click',(e)=>{  
    console.log(e.target.textContent);  
})
```

결론 : this를 쓸거면 function, 안쓸거면 아예 화살표 함수로 통일