setInterval에서 this 사용 방법

1. 변수 할당을 통한 scope 이동
2. class Person {
3. constructor(name) {
4. this.name = name;
5. }
6. print() {
7. var setInterval = window.setInterval;
8. setInterval(function() {
9. console.log(this.name);
10. }, 1000);
11. }
12. }
13. const NAME = "Danuel";
14. const person = new Person(NAME);

person.print();

변수로 할당해서 this를 window에서 Person로 가리키게 하는 원리입니다.

가장 간단한 방법이지만 JavaScript를 조금 더 이해하고 있어야 의도를 알 수 있다는 이슈가 있습니다. JavaScript에 익숙하지 않거나 다른 언어를 하던 분이라면 무의미한 코드라고 생각하고 var setInterval = window.setInterval 을 지울 수도 있습니다.

1. this를 가리키는 변수 생성
2. class Person {
3. constructor(name) {
4. this.name = name;
5. }
6. print() {
7. var self = this;
8. setInterval(function() {
9. console.log(self.name);
10. }, 1000);
11. }
12. }
13. const NAME = "Danuel";
14. const person = new Person(NAME);

person.print();

setInterval 바깥에서 this를 가리키는 변수를 생성하고, 그 변수를 사용하는 방법입니다.

1번째 방법과는 다르게, JavaScript에 대한 이해가 부족하더라도 코드를 읽으면서 자연스럽게 이해할 수 있습니다. 만약 이해를 못 하더라도 무엇인가 의도가 있을 것이라 예상할 수 있습니다.

1. Arrow Function 사용
2. class Person {
3. constructor(name) {
4. this.name = name;
5. }
6. print() {
7. setInterval(() => {
8. console.log(this.name);
9. }, 1000);
10. }
11. }
12. const NAME = "Danuel";
13. const person = new Person(NAME);

person.print();

자신의 상위 scope를 가리키는 Arrow Function의 특성을 이용하는 방법입니다.

앞서 소개한 방법과 다르게 JavaScript의 동작을 이해해야 할 필요가 없다는 장점이 있지만, 인터넷 익스플로러(Internet Explorer)는 지원하지 않는 문법이라는 이슈가 있습니다. 따라서, 지원해야 하는 브라우저가 이 문법을 지원하는지 먼저 살펴보아야 합니다.

출처: [this in setInterval (velog.io)](https://velog.io/@public_danuel/this-in-setinterval)