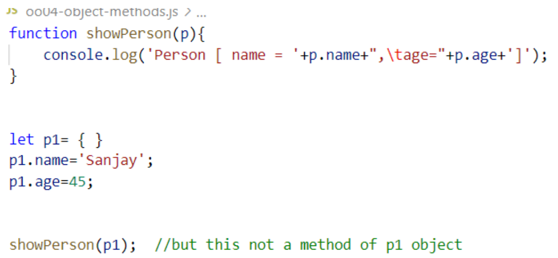
Object Behavior

Thursday, March 10, 2022 5:22 PM

- Our objects also need to implement functionalities using methods
- We can apply methods to implement them

Approach #1 — using global method to access an object

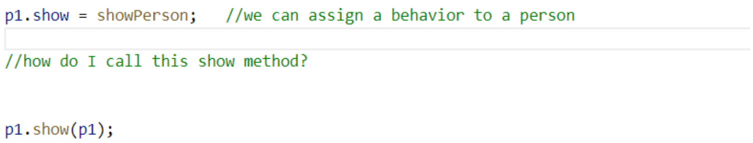


```
JS oo04-object-methods.js > ...
function showPerson(p){
    console.log('Person [ name = '+p.name+",\tage="+p.age+']');
}


let p1= { }
p1.name='Sanjay';
p1.age=45;


showPerson(p1);  //but this not a method of p1 object
```

- showPerson is not a part of the person object.

Approach #2 — we can assign a function as a property to the object

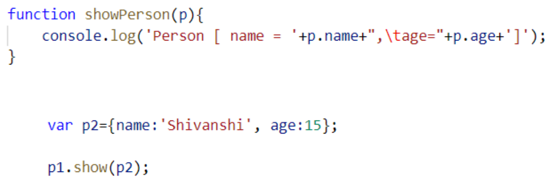
1. Functions are objects like number and string
2. We can assign any value to the property of an object



```
p1.show = showPerson;   //we can assign a behavior to a person

//how do I call this show method?


p1.show(p1);
```

Problem

- Why is 'p1' used twice in the show call?
- Which object will be displayed if we call



```
function showPerson(p){
    console.log('Person [ name = '+p.name+",\tage="+p.age+']');
}


    var p2={name:'Shivanshi', age:15};

    p1.show(p2);
```

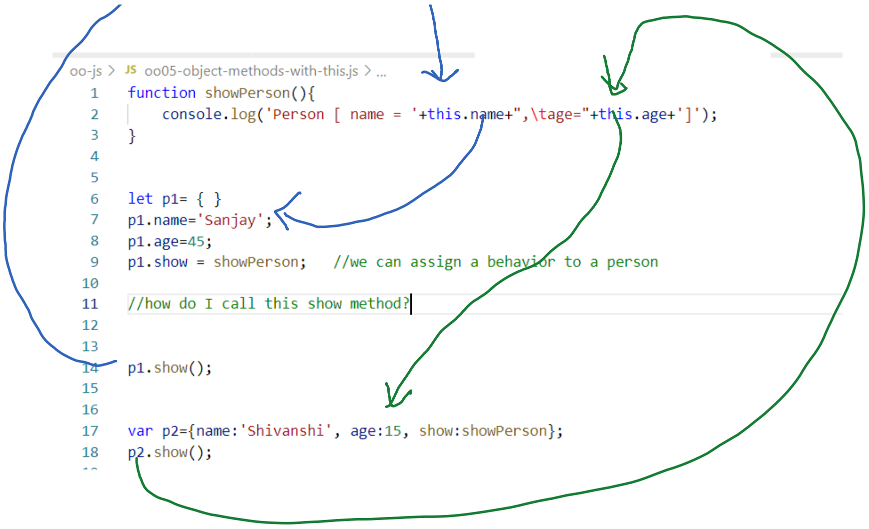
What is the role of p1?

- p1 just a cosmetic here. It has no role.
- p1 is NOT accessible within the function.

'this' reference

- JavaScript passes an hidden reference to every javascript function called 'this'
- This refers to the invoking object
    - Invoking object for a method is the object that comes before the dot.

```
function  show(){

    console.log(     this.name   );

}


var  person = new Object();

person.name= 'Sanjay'

person.show = show;


person.show();
```

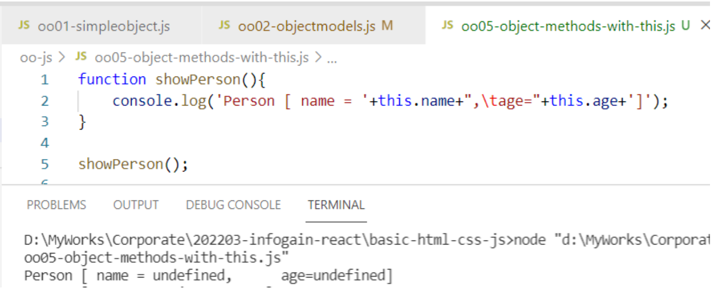


```
function showPerson(){
    console.log('Person [ name = '+this.name+",\tage="+this.age+']');
}


let p1= { }
p1.name='Sanjay';
p1.age=45;
p1.show = showPerson;   //we can assign a behavior to a person

//how do I call this show method?


p1.show();


var p2={name:'Shivanshi', age:15, show:showPerson};
p2.show();
```

Challenge Question! What will following code do

```
function showPerson(){
    console.log('Person [ name = '+this.name+",\tage="+this.age+']');
}

showPerson();
```

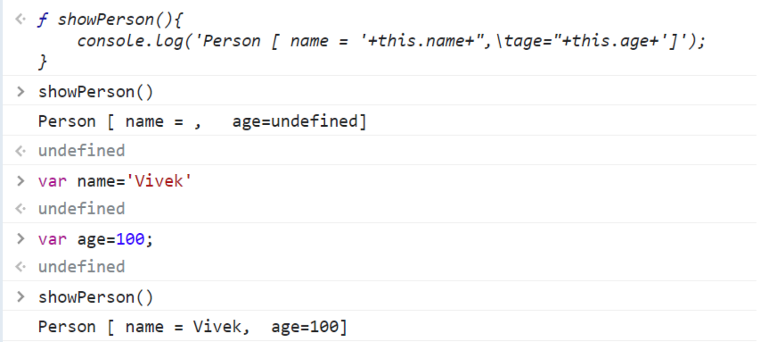
What is the value of 'this' here?


JS oo01-simpleobject.js
JS oo02-objectmodels.js M
JS oo05-object-methods-with-this.js U ×
oo-js > JS oo05-object-methods-with-this.js > ...
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL

```
function showPerson(){
    console.log('Person [ name = '+this.name+",\tage="+this.age+']');
}

showPerson();
```

```
D:\MyWorks\Corporate\202203-infogain-react\basic-html-css-js>node "d:\MyWorks\Corporat
oo05-object-methods-with-this.js"
Person [ name = undefined,      age=undefined]
```

- name , age are undefined.
- But 'this' is not undefined.
    - If 'this' were undefined I would get error like 'this' is not defined.



```
< ƒ showPerson(){
      console.log('Person [ name = '+this.name+",\tage="+this.age+']');
  }
> showPerson()
  Person [ name = ,     age=undefined]
< undefined
> var name='Vivek'
< undefined
> var age=100;
< undefined
> showPerson()
  Person [ name = Vivek,   age=100]
```

- Why is it showing global variables name and age?

Remember

Browser based applicatic

- In a browser based applic variable and method belc object
- When you write
    - **showPerson()**
- You are actually writing
    - **window.showPers**
- Here 'this' will point to 'w
- 'this' is never null

Node Application

- This is considered not nul
- The object belongs global
- But there is not global ob

- this.name will NOT refer t variable

How do we create Object with property and methods