

JS - this

Monday, May 1, 2023 5:54 PM

1. Globalna razina - bilo gdje izvan funkcije

this je window objekt

```
> this
< ▶ Window {window: Window, self: Window, document: document, name: '', Location: Location, ...}
> |
```

2. Unutar funkcije

this je objekt koji poziva funkciju

Primjer 1:

```
> function pjevaj() {
  return this;
}
< undefined

> const ptica = {
  ime: 'kos',
  pjevaj: pjevaj
};
< undefined

> ptica.pjevaj();
< ▶ {ime: 'kos', pjevaj: f}
>
```

Primjer 2:

```
> function intro() {
  return `Ja sam ${this.ime}`;
}
< undefined

> const ivica = {
  ime: 'Ivica',
  intro: intro
};
< undefined

> const marica = {
  ime: 'Marica',
  intro: intro
};
< undefined

> ivica.intro();
< 'Ja sam Ivica'

> marica.intro();
< 'Ja sam Marica'
>
```

3. Binding - dodavanje funkcije objektu ako je objekt nepromjenjiv

```

> function intro() {
    return `Ja sam ${this.ime}`;
}
< undefined

> const ivica_nepromjenjivi = {
    ime: 'Ivica Nepromjenjivi'
};
< undefined

> intro.bind(ivica_nepromjenjivi);
< f intro() {
    return `Ja sam ${this.ime}`;
} ⓘ

> intro.bind(ivica_nepromjenjivi)();
< 'Ja sam Ivica Nepromjenjivi'

> |

```

intro.bind(ivica_nepromjenjivi); -->> argument funkcije bind (ivica_nepromjenjivi) će biti vrijednost this unutar funkcije intro

Konstruktor funkcije automatski kreiraju bind između objekta koji se kreira i this unutar funkcije - this unutar konstruktor funkcije je objekt koji se kreira sa new

Primjer:

```

> function Osoba(n) {
    this.ime = n,
    this.govor = function() {
        console.log(this);
    }
}
< undefined

> const ivan = new Osoba('Ivan');
< undefined

> ivan.govor();
▶ Osoba {ime: 'Ivan', govor: f}

< undefined

>

```

Iznimka: unutar Callback funkcija (funkcije koje primaju funkciju kao argument) this je window, zato što se callback funkcije izvršavaju u odvojenom kontekstu

```

> function Osoba(n) {
    this.ime = n;
    this.govor = function() {
        console.log(this);
    };
    setTimeout(function(){
        console.log(this);
    }, 2000);
}
< undefined
> const ivan = new Osoba('Ivan');
< undefined
▶ Window {window: Window, self: Window, document: document, name: '', Location: Location, ...}
>

```

To se može riješiti na dva načina:

Prvi način - pomoću bind

```

> function Osoba(n) {
  this.ime = n;
  this.govor = function() {
    console.log(this);
  };
  setTimeout(function(){
    console.log(this);
  }.bind(this), 2000);
}
< undefined
> const ivan = new Osoba('Ivan');
< undefined
  ► Osoba {ime: 'Ivan', govor: f}
> |

```

Drugi način - pomoću arrow funkcije

```

> function Osoba(n) {
  this.ime = n;
  this.govor = function() {
    console.log(this);
  };
  setTimeout(() => {
    console.log(this);
  }, 2000);
}
< undefined
> const ivan = new Osoba('Ivan');
< undefined
  ► Osoba {ime: 'Ivan', govor: f}
>

```