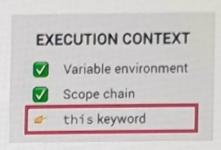# HOW THE THIS KEYWORD WORKS

👉 **this keyword/variable:** Special variable that is created for every execution context (every function). Takes the value of (points to) the "owner" of the function in which the this keyword is used.

🔥 this is **NOT** static. It depends on **how** the function is called, and its value is only assigned when the function **is actually called**.

EXECUTION CONTEXT

- ✅ Variable environment
- ✅ Scope chain
- 👉 this keyword

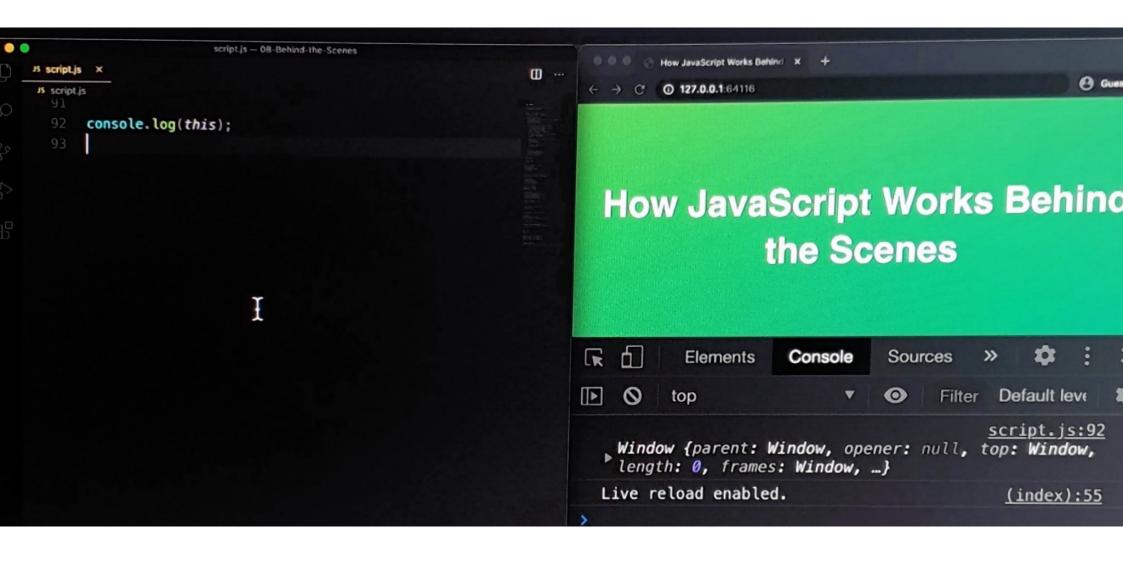👉 Method example:

```
const jonas = {
    name: 'Jonas',
    year: 1989,
    calcAge: function() {
        return 2037 - this.year
    }
};
jonas.calcAge(); // 48
```

calcAge is method   jonas   1989

Way better than using jonas.year!

Method 👉 this = <Object that is calling the method>

Simple function call 👉 this = undefined   In strict mode! Otherwise: window (in the browser)

Don't get own this →

Arrow functions 👉 this = <this of surrounding function (lexical this)>

Event listener 👉 this = <DOM element that the handler is attached to>

new, call, apply, bind 👉 <Later in the course... ⏳>

🔥 this does **NOT** point to the function itself, and also **NOT** the its variable environment!

JS script.js ✕

JS script.js

```
91
92    console.log(this);
93    |
```

How JavaScript Works Behind ✕ +

← → C ⓘ 127.0.0.1:64116

How JavaScript Works Behind the Scenes

Elements **Console** Sources »

top

Filter Default leve

script.js:92

▶ Window {parent: Window, opener: null, top: Window, length: 0, frames: Window, …}

Live reload enabled. (index):55
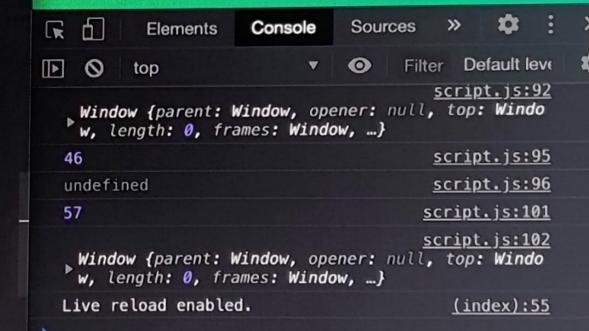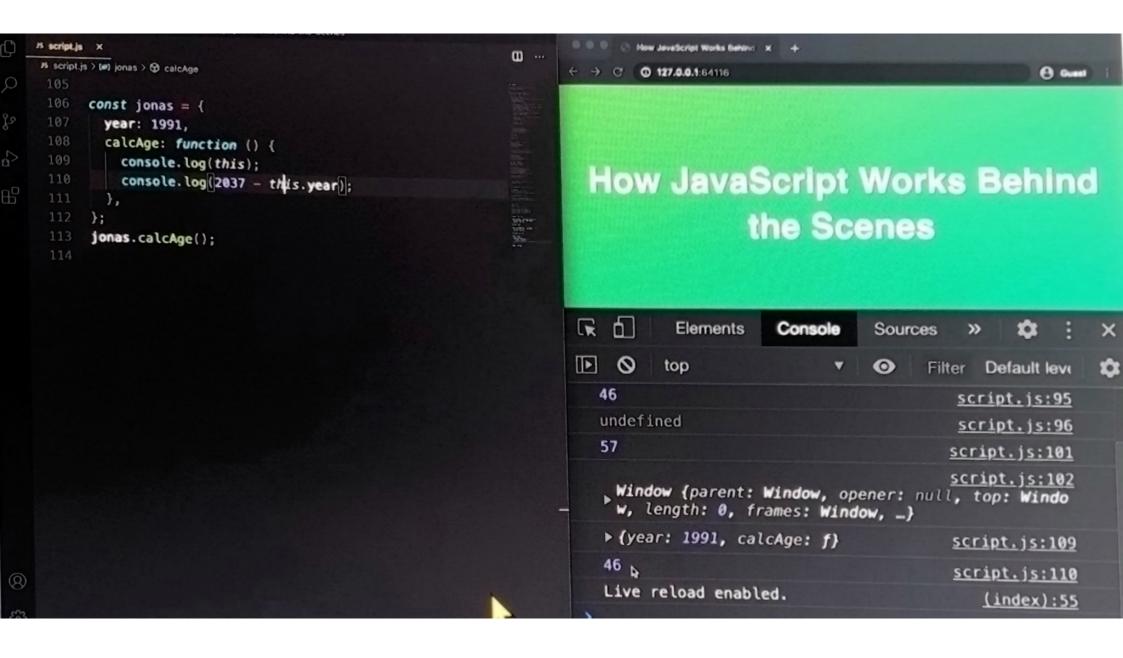
>

```js
91
92      console.log(this);
93
94      const calcAge = function (birthYear) {
95        console.log(2037 - birthYear);
96        console.log(this);
97      };
98      calcAge(1991);
99
```

How JavaScript Works Behind the Scenes

127.0.0.1:64116

Elements | Console | Sources | »

top | Filter | Default leve

```
► Window {parent: Window, opener: null, top: Window,      script.js:92
    length: 0, frames: Window, …}

46                                                         script.js:95

undefined                                                  script.js:96

Live reload enabled.                                       (index):55
```

```javascript
91
92    console.log(this);
93
94    const calcAge = function (birthYear) {
95      console.log(2037 - birthYear);
96      console.log(this);
97    };
98    calcAge(1991);
99
100   const calcAgeArrow = birthYear => {
101     console.log(2037 - birthYear);
102     console.log(this);
103   };
104   calcAgeArrow(1980);
105
```

How JavaScript Works Behind the Scenes

Elements | **Console** | Sources | »

top ▼ 👁 Filter Default level

```
                                              script.js:92
  ▸ Window {parent: Window, opener: null, top: Windo
    w, length: 0, frames: Window, …}

  46                                          script.js:95

  undefined                                   script.js:96

  57                                          script.js:101

                                             script.js:102
  ▸ Window {parent: Window, opener: null, top: Windo
    w, length: 0, frames: Window, …}

  Live reload enabled.                         (index):55
```

```
112   };
113   jonas.calcAge();
114
115   const matilda = {
116     year: 2017,
117   };
118
119   matilda.calcAge = jonas.calcAge;
120
```

How JavaScript Works Behind the Scenes

Elements   **Console**   Sources   »   ⚙   ⋮

top   ▾   👁   Filter   Default leve

```
w, length: 0, frames: Window, …}
▶ {year: 1991, calcAge: f}              script.js:109
46                                      script.js:110
Live reload enabled.                    (index):55
> matilda
‹· ▶ {year: 2017, calcAge: f}
> jonas
‹· ▶ {year: 1991, calcAge: f}
>
```

```js
104    calcAgeArrow(1980);
105
106    const jonas = {
107      year: 1991,
108      calcAge: function () {
109        console.log(this);
110        console.log(2037 - this.year);
111      },
112    };
113    jonas.calcAge();
114
115    const matilda = {
116      year: 2017,
117    };
118
119    matilda.calcAge = jonas.calcAge;
120    matilda.calcAge();
121
```

How JavaScript Works Behind the Scenes

Elements **Console** Sources »

top ▾   Filter Default leve

46	script.js:95
57	script.js:101
▸ {year: 1991, calcAge: f}	script.js:109
46	script.js:110
▸ {year: 2017, calcAge: f}	script.js:109
20	script.js:110
Live reload enabled.	(index):55

>

```
104    calcAgeArrow(1980);
105
106    const jonas = {              this: {
107      year: 1991,                   year: number;
108      calcAge: function ()          calcAge: () => void;
109        console.log(this);        }
110        console.log(2037 - this.year);
111      },
112    };
113    jonas.calcAge();
114
115    const matilda = {
116      year: 2017,
117    };
118
119    matilda.calcAge = jonas.calcAge;
120    matilda.calcAge();
121
122    const f = jonas.calcAge;
123    f();
124
```

How JavaScript Works Behind the Scenes

Console output:

46	script.js:110
▶ {year: 2017, calcAge: f}	script.js:109
20	script.js:110
undefined	script.js:109
⊗ ▶ Uncaught TypeError: Cannot read property 'year' of undefined	script.js:110
at calcAge (script.js:110)	
at script.js:123	
Live reload enabled.	(index):55

```js
127
128    const jonas = {
129      firstName: 'Jonas',
130      year: 1991,
131      calcAge: function () {
132        console.log(this);
133        console.log(2037 - this.year);
134      },
135
136      greet: () => console.log(`Hey ${this.firstName}`),
137    };
138    jonas.greet();
139    console.log(this.firstName);
140
```

How JavaScript Works Behind the Scenes

Elements | **Console** | Sources | »

top

Filter   Default leve

Hey undefined   script.js:136
undefined   script.js:139
Live reload enabled.   (index):55
>

```javascript
120  */
127
128    var firstName = 'Matilda';
129
130    const jonas = {
131      firstName: 'Jonas',
132      year: 1991,
133      calcAge: function () {
134        console.log(this);
135        console.log(2037 - this.year);
136      },
137      const jonas: {
138          firstName: string;
139          year: number;
140          calcAge: () => void;  s.firstName}`);
141          greet: () => void;
142      }
143    jonas.greet();
144
```


How JavaScript Works Behind the Scenes

| Elements | **Console** | Sources | » |

top ▼ Filter Default leve

```
                                                              script.js:139
  ▶ Window {parent: Window, opener: null, top: Window,
    length: 0, frames: Window, …}

  Hey Matilda                                                 script.js:140
  Live reload enabled.                                        (index):55
  >
```

```javascript
133    calcAge: function () {
134      // console.log(this);
135      console.log(2037 - this.year);
136
137      // Solution 1
138      // const self = this; // self or that
139      // const isMillenial = function () {
140      //    console.log(self);
141      //    console.log(self.year >= 1981 && self.year
         <= 1996);
142      // };
143
144      // Solution 2
145      const isMillenial = () => {
146        console.log(this);
147        console.log(this.year >= 1981 && this.year <=
         1996);
148      };
149      isMillenial();
150    },
151
152    greet: () => {
153      console.log(this);
154      console.log(`Hey ${this.firstName}`);
155    },
156  };
157  jonas.greet();
158  jonas.calcAge();
159
```

Browser: 127.0.0.1:64116 — Guest

How JavaScript Works Behind the Scenes

Console — Elements | Console | Sources | »

top | Filter | Default leve

```
                                                    script.js:153
▶ Window {parent: Window, opener: null, top: Windo
  w, length: 0, frames: Window, …}
Hey undefined                                       script.js:154
46                                                  script.js:135
                                                    script.js:146
▶ {firstName: "Jonas", year: 1991, calcAge: f, gree
  t: f}
true                                                script.js:147
Live reload enabled.                                (index):55
```