

Objekter

Objekter

- ❖ Alt i JavaScript er objekter – pånær de primitive typer
- ❖ Et objekt er en dynamisk samling af properties
- ❖ En property har et navn og en værdi
- ❖ Et navn kan være et JavaScript navn eller en string
- ❖ En værdi kan være af en vilkårlig type
- ❖ Objekter kan skrives som en literal

```
// literal.js  
let o = {id: 123, '3 tal': [7, 9, 13]};
```

Properties

- ♣ En property med et JavaScript navn kan tilgås med . eller [] notation
- ♣ En property med et andet navn kan kun tilgås med [] notation

```
// properties.js
let o = {a: 1, 'b': 2, '3': 3};

console.log(o.a); // => 1
console.log(o['a']); // => 1
// console.log(o[a]); // => ReferenceError: a is not defined

console.log(o.b); // => 2
console.log(o['b']); // => 2

console.log(o['3']); // => 3
console.log(o[3]); // => 3
// console.log(o.3); // => Compile error
```

CRUD på objekter

```
// crud.js
let a = 1;
let o = {a, 'b': 2, '3': 3}; // ny property notation med variabel
// let o = {a: a, 'b': 2, '3': 3};
o.c = 4;
console.log(o.b); // => 2
o['3'] = 'III';
delete o.b;
console.log(o); // => { '3': 'III', a: 1, c: 4 }
console.log(o.d); // => undefined
console.log('d' in o); // => false
```

Iteration over properties

```
// iteration.js
let o = {a: 1, 'b': 2, '3': 3};
for (let n in o)
    console.log(n + ": " + o[n]);

// => 3: 3
// => a: 1
// => b: 2
```

Metoder

- ♣ En funktion kaldes en metode, hvis den er værdien af en property
- ♣ En metode kan kaldes med . eller [] notation
- ♣ this refererer til en metodes objekt

```
// metode.js
let x = 1;
let o = {x: 2, m(){return this.x;}}; // ny metode notation
// let o = {x: 2, m: function(){return this.x;}};
console.log(o.m()); // => 2
console.log(o["m"]()); // => 2
let f = o.m;
console.log(f()); // => undefined
let g = function(){return x};
console.log(g()); // => 1
o.m = g;
console.log(o.m()); // => 1
```

Wrapper Objekter

♣ Number, String og Boolean
er wrapper objekter til de
simple typer

```
// wrapper.js
let i = 123;
console.log(i.toExponential()); // => 1.23e+2
let s = "test";
console.log(s.p); // => undefined
console.log(s.length); // => 4
console.log(s[2]); // => s
console.log(s.toUpperCase()); // => TEST
let b = true;
console.log(b.toString()); // => true
```