

A blue parallelogram and a light green parallelogram are positioned in the upper left corner of the slide. The blue shape is partially behind the green one. Both shapes are oriented diagonally, with their longer sides running from the top-left towards the bottom-right.

Övning ger
färdighet



Agenda

- Läxa genomgång
- Repetition
 - Funktioner
 - Arrayer
 - Object
 - DOM manipulation
- Övningar
 - Clock (exempel)
 - Array cardio
 - Colors
 - Tictactoe (ny version)



Functions

```
function logIt(name) {  
  console.log(name);  
}
```

logit (definition)

name är en parameter (en variabel) som är tom i definition

kodblock är det maskineri som väntar på att köras.

```
function logIt(name) {  
  console.log(name);  
}
```



Functions

```
function logIt(name) {  
  console.log(name);  
}  
  
logIt("Kenny"); // Kenny
```

logit (exekvering)

Vi kallar på funktionen med argument "Kenny" till parameter name



Functions

```
function logIt2() {  
  console.log("The second one");  
}
```

```
logIt2(); // The second one
```

Funktioner utan parameter

logit2 tar inga argument, bara tom parantes för att exekvera



Functions

```
function test() {}  
  
const x = test();  
console.log(x); // undefined
```

Funktioner utan return

Eftersom test inte returnerar något får vi undefined i x



Functions

```
function test() {  
  return true;  
}  
  
const x = test();  
console.log(x); // true
```

Funktioner med return

Men om vi returnerar något kan vi samla ihop det i variabel



Fråga

```
function test(nr) {  
  console.log("Hej");  
  return nr * 20;  
}
```

```
const x = test(1);  
console.log(x); // Vad ser vi i konsoll?
```




Fråga

```
function test() {  
  return true;  
  return false;  
}  
  
console.log(test()); // ???
```



Fråga

```
function isTerrorist(person) {  
  if (person === "Brejvik") {  
    return true;  
  } else if (person === "Binladen") {  
    return true;  
  } else if (person === "Netanyahu") {  
    return true;  
  } else if (person === "Haniyeh") {  
    return true;  
  } else {  
    return false;  
  }  
}  
  
console.log(isTerrorist("Brejvik")); // ???  
console.log(isTerrorist("Palme")); // ???
```



Object

Ett objekt består av block med properties

prop: value,

```
const person = {  
  name: "Ulf",  
  age: 45,  
  male: true,  
};
```



Object

Vi kan hämta och ändra i objekt properties

```
console.log(person.name); // Ulf
person.age++;
console.log(person.age); // 46
person.male = false;
console.log(person.male); // false
```

```
const person = {
  name: "Ulf",
  age: 45,
  male: true,
};
```



Object

printName tar ett objekt som parameter

(räknar med property "name")

```
function printName(person) {  
  console.log(person.name);  
}
```

```
printName({ name: "Ulf", age: 45 }); // Ulf
```



Object

Uppgift: skriv en funktion "printAge" som tar ett person objekt och printar ut age istället.

```
function printName(person) {  
  console.log(person.name);  
}
```

```
printName({ name: "Ulf", age: 45 }); // Ulf
```



typeof

```
typeof 1; // "number"  
typeof false; // "boolean"  
typeof "hej"; // "string"  
typeof undefined; // "undefined"  
typeof [1, 2]; // "object"  
typeof { name: "ulf", age: 45 }; // "object"  
typeof function test() {}; // "function"
```



Fråga

```
function countAllNumbers(numbers) {  
  let count = 0;  
  for (let i = 0; i < numbers.length; i++) {  
    if (typeof numbers[i] === "number") {  
      count++;  
    } else {  
      return count; // avbryt loop  
    }  
  }  
  return count;  
}
```

```
const ns1 = countAllNumbers([1, 2, 3, "hej", 4, 5]); // ?  
const ns2 = countAllNumbers([1, 2, 5, 7, "whaat", 7]); // ?
```




Fråga

```
function countAllNumbers(numbers) {  
  let count = 0;  
  for (let i = 0; i < numbers.length; i++) {  
    if (typeof numbers[i] === "number") {  
      count++;  
    } else {  
      return count; // avbryt loop  
    }  
  }  
  return count;  
}  
  
const ns1 = countAllNumbers([1, 2, 3, "hej", 4, 5]); // ?  
const ns2 = countAllNumbers([1, 2, 5, 7, "whaat", 7]); // ?
```

Uppgift: Skapa funktion (isAllNames) som tar en array (persons) med strängar (["Alice", "Bertil"]) och kontrollera att alla element i (persons) är strängar. Om dom alla inte alla är strängar returnera false, annars true



DOM manipulation

`document.querySelector(css selector)`

```
<body>
  <div class="clock">
    <div class="clock-face">
      <div class="hand hour-hand"></div>
      <div class="hand min-hand"></div>
      <div class="hand second-hand"></div>
    </div>
  </div>

  <script>
    const secondHand = document.querySelector(".second-hand");
    const minsHand = document.querySelector(".min-hand");
    const hourHand = document.querySelector(".hour-hand");
  </script>
</body>
```



DOM manipulation

`document.createElement(tag name)`

```
<div class="clock">
  <div class="clock-face">
    <div class="hand hour-hand"></div>
    <div class="hand min-hand"></div>
    <div class="hand second-hand"></div>
  </div>
</div>

<script>
  const para = document.createElement("p");
  const text = document.createTextNode("Hejsan du!");
  para.appendChild(text); // <p>Hejsan du!</p>

  const clock = document.querySelector("clock");
  clock.appendChild(para);
  // <div class="clock">
  //   ...
  //   <p>Hejsan du!</p>
  // </div>
</script>
```



DOM manipulation

`element.innerHTML = new content;`

```
const para = document.createElement("p");
const text = document.createTextNode("Hejsan du!");
para.appendChild(text); // <p>Hejsan du!</p>

const clock = document.querySelector("clock");
clock.innerHTML = para;
// <div class="clock">
//   <p>Hejsan du!</p>
// </div>
```



DOM manipulation

```
const hour = document.querySelector(".hour-hand");

hour.classList.remove("hand");
// <div class="hour-hand"></div>

hour.classList.add("large");
// <div class="hour-hand large"></div>

hour.style.fontSize = "2em";
// <div style="font-size: 2em;" class="hour-hand large"></div>
```



DOM manipulation

`element.addEventListener(event, function)`

```
const para = document.createElement("p");
const text = document.createTextNode("Hejsan du!");
para.appendChild(text); // <p>Hejsan du!</p>

const clock = document.querySelector("clock");
clock.innerHTML = para;
// <div class="clock">
//   <p>Hejsan du!</p>
// </div>
```

Exempel (clock.html)



Array cardio (array-cardio.html)



```
[🐱, 🍌, 🐔, 🥦].map(cook) ⇒ [🍔, 🍗, 🍖, 🍗]  
[🍔, 🍗, 🍖, 🍗].filter(isVegetarian) ⇒ [🍗, 🍗]  
[🍔, 🍗, 🍖, 🍗].reduce(eat) ⇒ 🍌
```


Array cardio (array-cardio.html)

```
const numbers = [1, 2, 3, 4, 5];

numbers.forEach(function(number) {
  console.log(number);
});

// You will see 1, 2, 3, 4, 5 on separate lines of the console
```

```
const numbers = [2, 10, 3];

const doubled = numbers.map(function(number) {
  return number * 2;
});

console.log(doubled); // [4, 20, 6]
```

```
const words = [
  'cat',
  'discombobulate',
  'dog',
  'antidisestablishmentarianism',
  'pusillanimous',
  'owl',
];

const threeLetterWords = words.filter(function(word) {
  return word.length === 3;
});

console.log(threeLetterWords); // ["cat", "dog", "owl"]
```



Colors (colors.html)

Vi ska leka lite med css och träna DOM manipulation



Tictactoe (tictactoe.html, tictactoe.css)

