

# Föreläsning 3

1ME322 Webbteknik 2

<https://wt2.enur.se>

***Rune Körnefors***

Loopar  
Array  
String  
HTML-element  
DOM

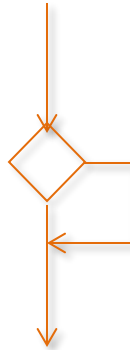
## Vanligt sekventiellt flöde

Programsatserna utförs en efter en i den ordning de står i programmet.



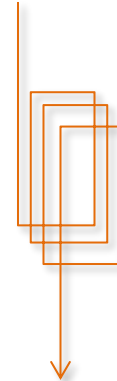
## Jämförelse (if-sats)

En eller flera programsatser utförs, beroende på resultatet av ett villkor.



## Loop

En eller flera programsatser utförs flera gånger.

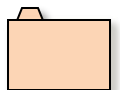


## Loopar

```
for (initiering; villkor; uppräkning) {  
    programsatser;  
}
```

```
while (villkor) {  
    programsatser;  
}
```

```
do {  
    programsatser;  
} while (villkor);
```



F3-ex1

## Variabel med ett värde

**country**

Sverige

## Variabel med en lista av värden

<b>countries</b>	
0	Sverige
1	Norge
2	Danmark
3	Tyskland
4	England
5	Polen
6	Frankrike
7	Spanien

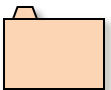
*index*      *värde*

**length**  
= antal värden

### Exempel på metoder:

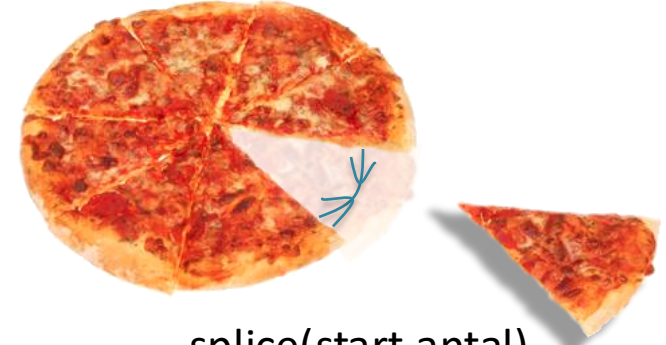
<b>push()</b>	– lägger till ett nytt värde sist i arrayen
<b>pop()</b>	– tar bort det sista värdet i arrayen
<b>includes()</b>	– true/false beroende på om värdet finns i arrayen
<b>indexOf()</b>	– ger index till första förekomsten av värdet
<b>lastIndexOf()</b>	– ger index till sista förekomsten av värdet
<b>slice()</b>	– kopia av en del av arrayen
<b>splice()</b>	– tar bort en del av arrayen
<b>sort()</b>	– sorterar arrayen
<b>join()</b>	– lägger ihop värdena till en textsträng

```
let countries = ["Sverige", "Norge", "Danmark", "Tyskland", "England", "Polen", "Frankrike", "Spanien"];
```





slice(start,slut)



splice(start,antal)

pizza	
0	A
1	B
2	C
3	D
4	E
5	F
6	G
7	H

pizza.slice(2,5)

C
D
E

pizza	
0	A
1	B
2	C
3	D
4	E
5	F
6	G
7	H

pizza.splice(2,3)

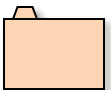
C
D
E

pizza	
0	A
1	B
2	F
3	G
4	H

pizza.slice(0) ger en kopia av hela arrayen.

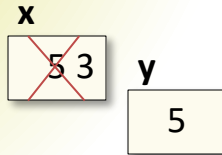
kopiera

ta bort och tryck ihop



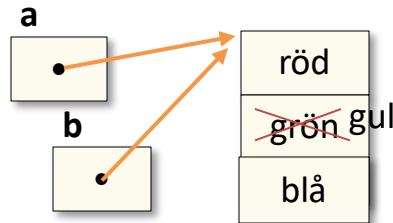
## Variabler med numeriska värden

```
x = 5;
y = x;
x = 3;
```



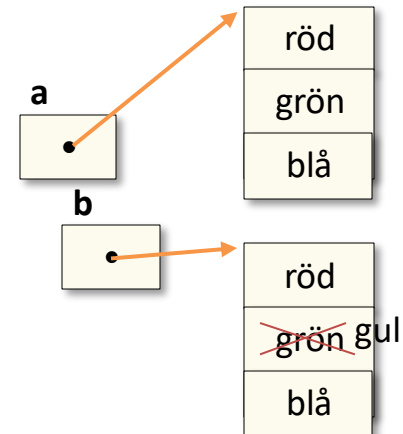
## Variabler med arrayer

```
a = ["röd", "grön", "blå"];
b = a;
b[1] = "gul";
console.log(a[1]);
→ gul
```

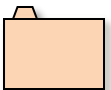


## Kopiera en array

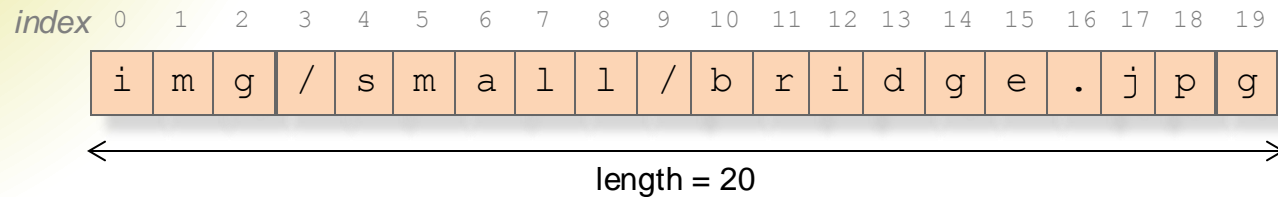
```
a = ["röd", "grön", "blå"];
b = a.slice(0);
b[1] = "gul";
console.log(a[1]);
→ grön
```



Alternativ:  
**b = [...a];**



Antag att variabeln *s* innehåller följande text  
*s* = "img/thumbs/bridge.jpg"

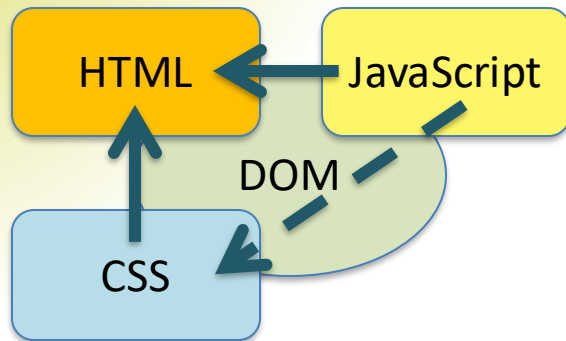


## Exempel på några egenskaper och funktioner

- `s.length` – antal tecken, **20**
- `s.includes(c)` – true eller false beroende på om tecknen finns, `s.includes("small")` ger **true**
- `s.indexOf(c)` – index för första förekomsten av tecknen i c, `s.indexOf("/")` ger **3**  
Om tecknen inte finns, ges värdet -1, `s.indexOf("large")` ger **-1**
- `s.lastIndexOf(c)` – index för sista förekomsten av tecknen i c, `s.lastIndexOf("/")` ger **9**
- `s.substring(i,k)` – en sträng med tecknen i position i till och med positionen före k, `s.substring(10,16)` ger **"bridge"**
- `s.substring(i)` – en sträng med tecknen från position i och resten av textsträngen, `s.substring(17)` ger **"jpg"**
- `s.split(c)` – delar upp strängen vid skiljetecknet i c och ger en array, `s.split("/")` ger **["img","small","bridge.jpg"]**

Kombinera detta:

Filnamnet i strängen kan erhållas med `s.substring(s.lastIndexOf("/") + 1)` som ger **"bridge.jpg"**



DOM — Document Object Model

Objekt för webbsidans innehåll, inklusive HTML-element, stil och händelser.

Standardiseras av W3C

I JavaScript refererar vi till HTML-element och kan förändra deras

- innehåll
- attribut
- stil
- händelser

HTML-kodens innehåll och struktur kan förändras genom

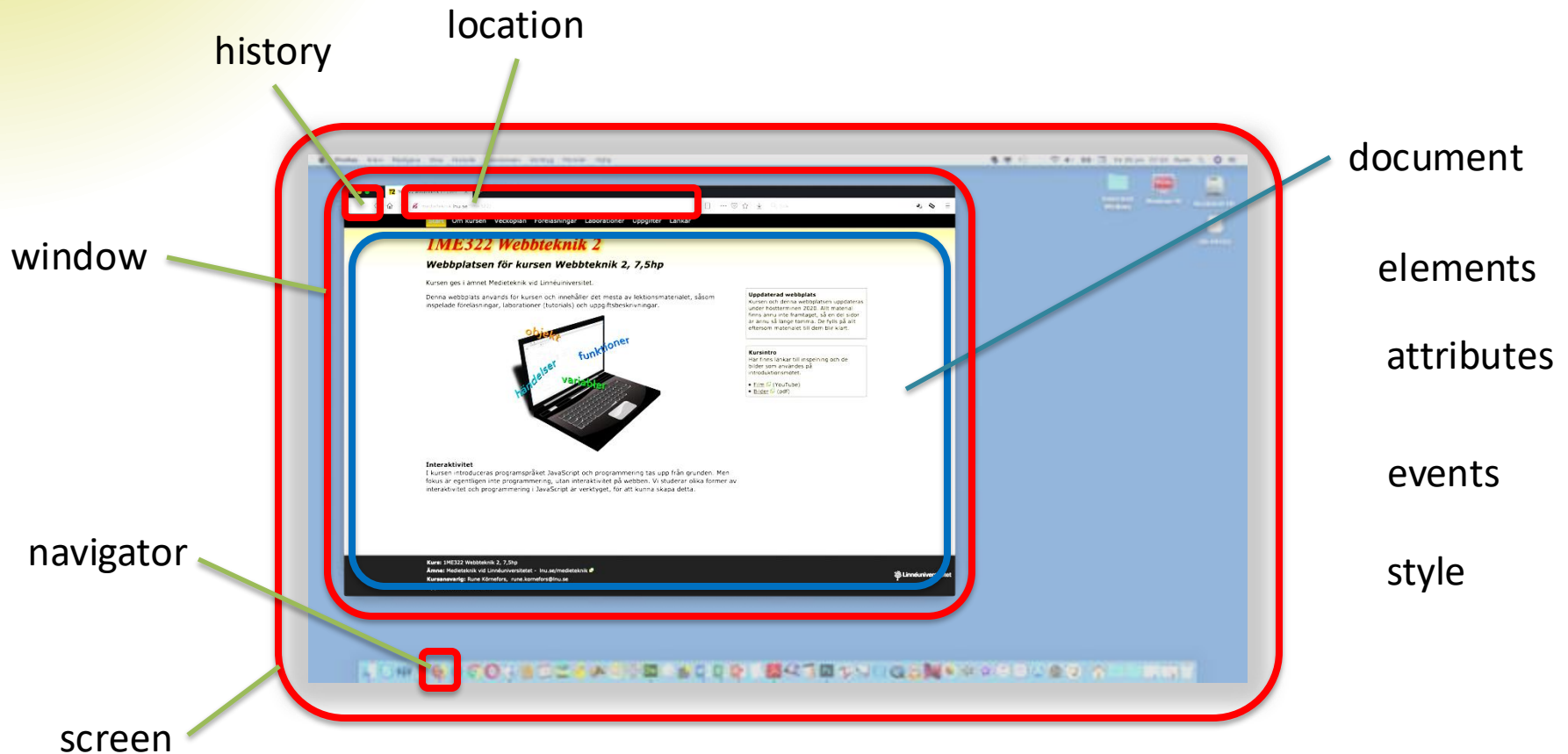
- att lägga in HTML-kod i elementets `.innerHTML`
- funktioner (metoder) för att skapa, ta bort och flytta element — tas upp i Webbteknik 3

## BOM

Browser Object Model

## DOM

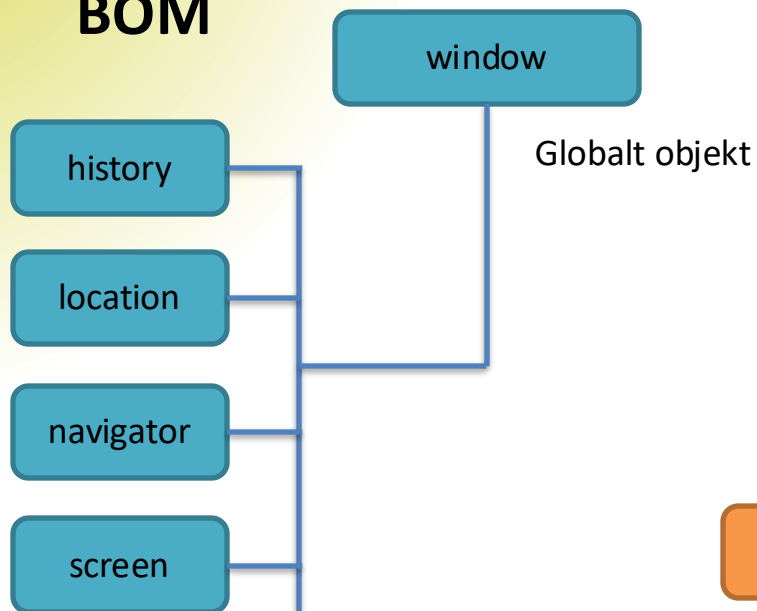
Document Object Model



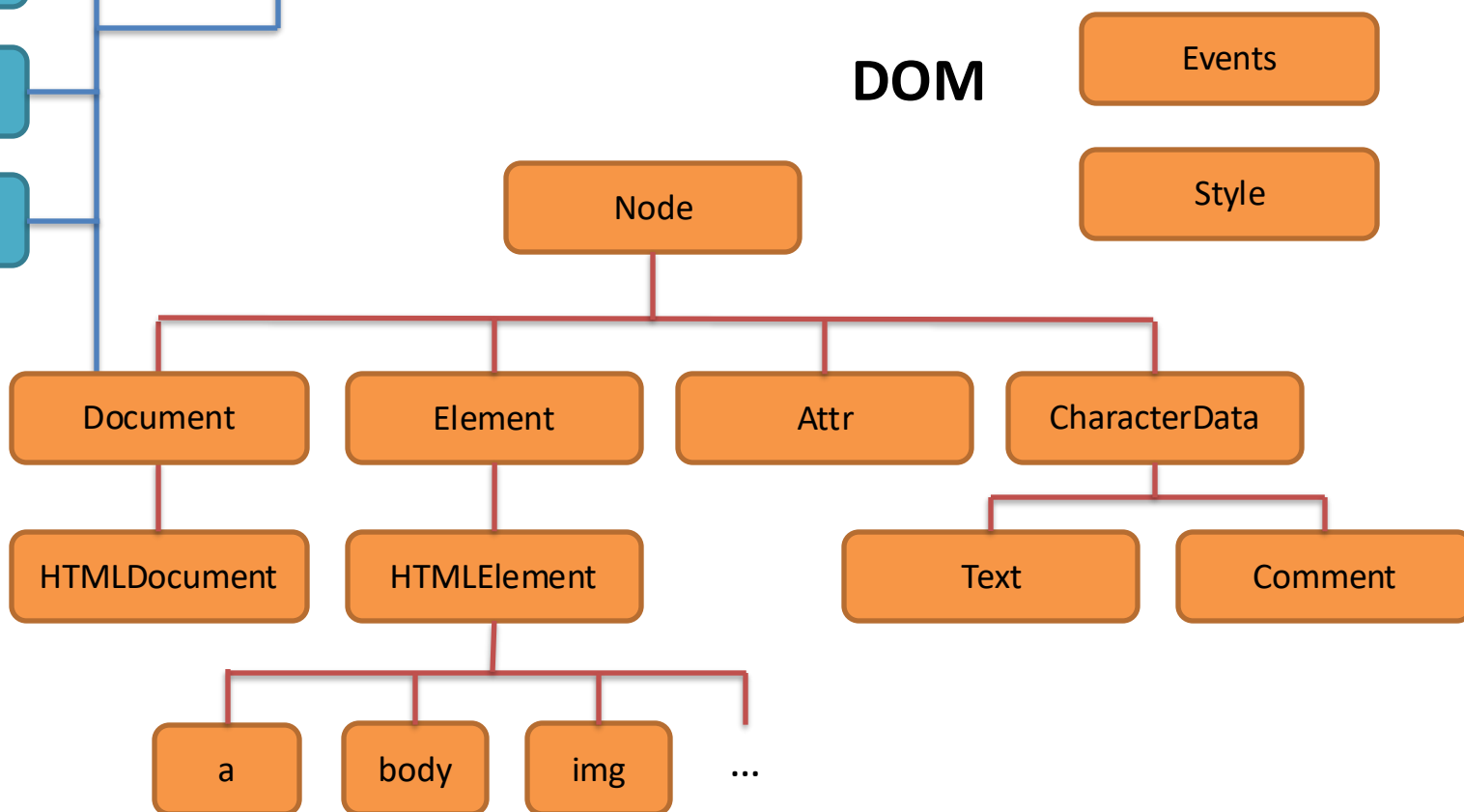


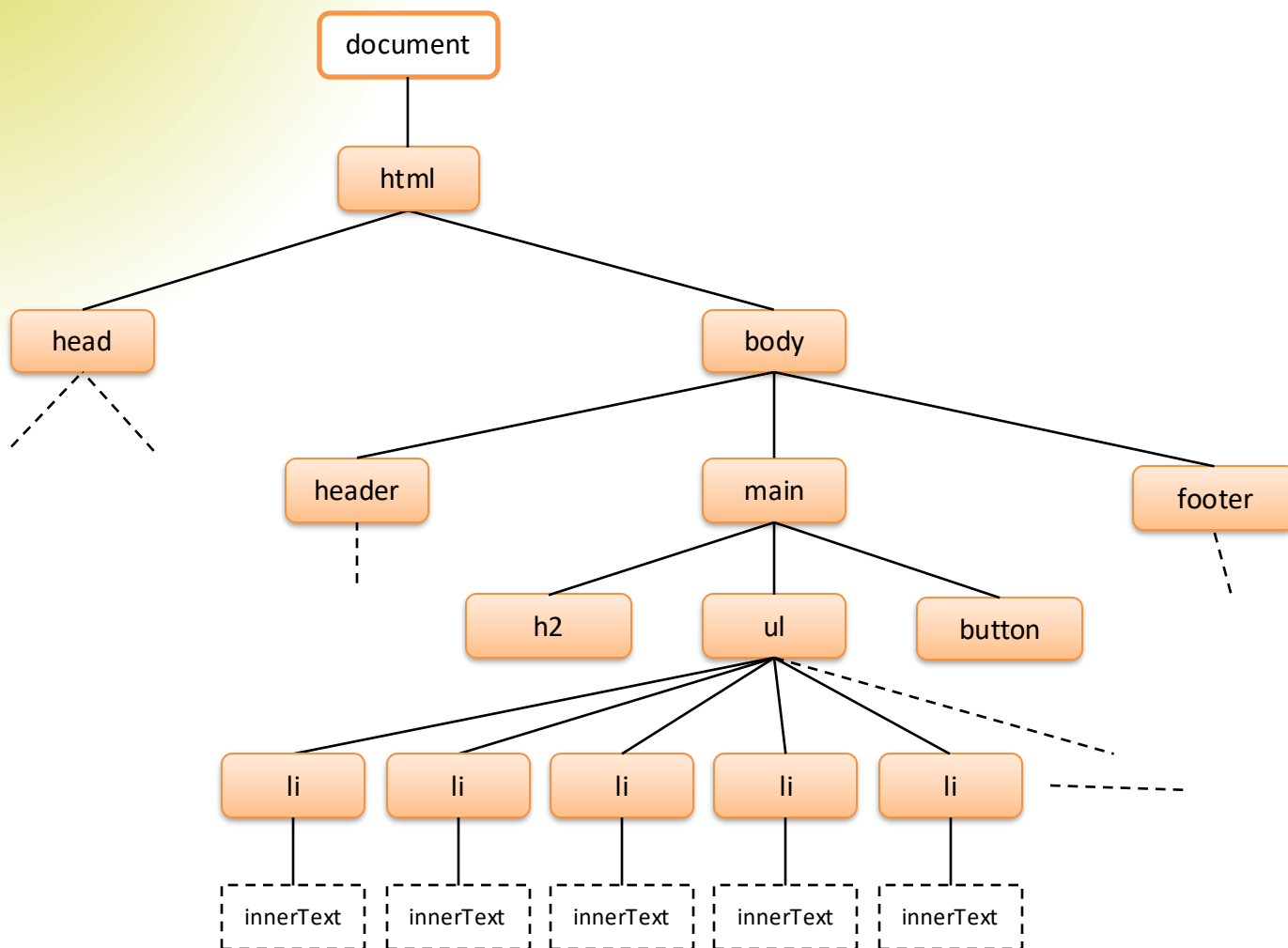
**BOM**

De vanligaste objekten

**DOM**

Objekttyper i DOM





*parentElement*



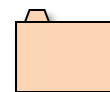
*"children"*

*querySelector()*  
*querySelectorAll()*

*previousElementSibling*



*nextElementSibling*



```
document.getElementById("abc")
```

```
document.getElementsByClassName("xyz")
```

```
document.getElementsByTagName("li")
```

```
document.getElementById("abc").getElementsByTagName("li")
```

```
document.querySelector("#abc")
```

```
document.querySelectorAll(".xyz")
```

```
document.querySelectorAll("li")
```

```
document.querySelectorAll("#abc li")
```

`getElementsByClassName("xyz")` → dynamisk array  
Byter man class i något av elementen, tas det bort ur arrayen

`querySelectorAll(".xyz")` → statisk array  
Alla element finns kvar i arrayen,  
även om man byter class i något element

`getElemetsByClassName()` och `getElementsByTagName()` ger en [HTMLCollection](#).  
`querySelectorAll()` ger en [NodeList](#).

Båda är en lista av HTML-element / noder.

Fungerar i princip som en Array, där t.ex. `length` kan användas, men inte alla metoder.

Konvertering till Array med:

```
let myElements = Array.from(document.querySelectorAll("..."));
```

Studera exempel och övningar på [js.enur.se](http://js.enur.se)

Exempel 2.10 - 2.13

- Loopar
  - for och while
  - break
- Array
  - Lista av värden
  - Indexering, length samt några olika metoder
- String (textsträng)
  - Sekvens av 0 eller flera tecken
  - Indexering, length samt några olika metoder
- Element
  - HTML-element och attribut – DOM, Document Object Model
  - `querySelector()` och `querySelectorAll()`
  - `parent` och `siblings`

Påbörja uppgift U1b