

Föreläsning 4

1ME322 Webbteknik 2

<https://wt2.enur.se>

Rune Körnefors

Slumptal
Datum och tid
Interaktivitet och gränssnitt
Händelser

`Math.random()`

→ ett tal mellan 0.000... och 0.999...

`Math.floor(n * Math.random())`

→ ett heltal mellan 0 och n-1

`Math.floor(n * Math.random()) + 1`

→ ett heltal mellan 1 och n

`Math.floor(41 * Math.random()) + 30`

→ ett heltal mellan 30 och 70

Slumpmässigt index till en array

`Math.floor(a.length * Math.random())` → ett heltal mellan 0 och högsta index i a

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

```
let r = Math.floor(countries.length * Math.random());  
randomCountry = countries[r];
```

```
let oldCountry = randomCountry;  
while (oldCountry == randomCountry) {  
  let r = Math.floor(countries.length * Math.random());  
  randomCountry = countries[r];  
}
```

```
let r = Math.floor(countries.length * Math.random());  
randomCountry = countries[r];  
countries.splice(r, 1);
```

Tre olika alternativ

```
let myDate = new Date();
```

```
let year = myDate.getFullYear();
```

```
let hour = myDate.getHours();
```

```
...
```

Mäta tid

programtid



```
let now = new Date();  
let startTime = now.getTime();
```

```
let now = new Date();  
let endTime = now.getTime();  
let timeElapsed = endTime - startTime;
```



Gränssnitt i Månbas Alpha
(Space: 1999) från 1975.



Gränssnitt i Star Trek
Voyager från 1995.



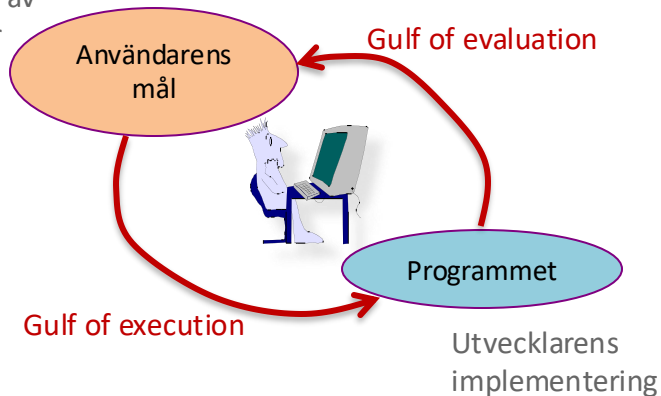
Gränssnitt i verkligheten idag

Från Webbteknik 1

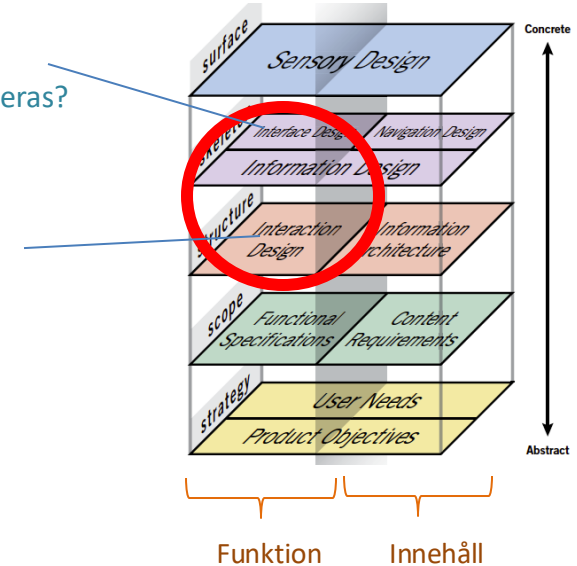
Hur ska gränssnittet se ut?
Vilka objekt ska användas?
Var ska delarna i gränssnittet placeras?

Vad ska användaren respektive
programmet göra?
Design av dialogen mellan
användare och program.

Användarens
uppfattning av
vad och hur
man gör.



Garretts designmodell



Utförandegap (Gulf of execution)

Avstånd mellan användarens mål och
möjlighet att uppnå dessa med
programmet.

Tolkningsgap (Gulf of evaluation)

Avstånd mellan implementering av
programmet och användarens mål.

Konceptuell design (Vad)

Bestäm syfte med programmet

Bestäm funktionalitet

Vad användaren ska göra, vad programmet ska göra

- Vad,
Uppgifter (tasks)



Fysisk design (Hur)

Bestäm gränssnitt

Visibility och affordance, stimuli samt feedback

Skapa gränssnittet i HTML och CSS

- Hur,
Handlingar (actions)



Strukturerar funktionaliteten i funktioner

Bestäm koppling (händelser) mellan gränssnitt och funktioner

Bestäm målrespons, verklig respons och programmets resultat

Bestäm vilken data som behövs

Variabler och parametrar

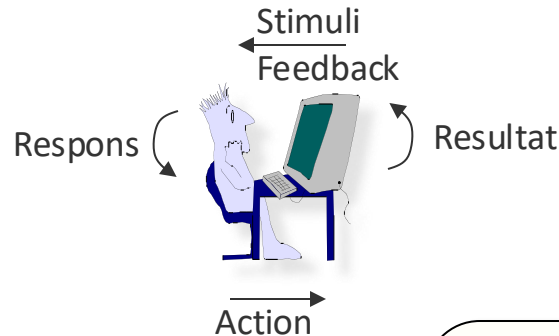
Skriv programkoden

Skapa programmet i små steg och testa efter varje nytt steg



"Give each action an immediate and obvious effect."

Norman, 1988, s.99



Användare

Program

Användaren styr vad som görs

3. Målrespons (förväntad respons)
Vad användaren kan göra för att uppnå målet.

4. Verklign respons
Användaren kan även göra något annat, vilket kan leda till andra resultat än målet.

Hantering av situationer där användaren kan göra "fel":

- Förhindra
- Rätta
- Återställa

Garrett, 2011, s.86

1. Stimuli
Visa att något kan göras.

2. Sättet att agera
Visa hur användaren kan agera
- klicka på knapp, välja i meny, ...

Aktivera användaren

Affordance

Datorns svar

5. Resultat
Vad som händer som direkt konsekvens av användarens handling.

Exempel:

- Lånekalkylen i ex1-1
- Spelet i uppgift U1b

Användarbaserade

- Musen (Mouse Events) — click, mouseenter, mouseleave, drag, drop, ...
- Tangentbordet (Keyboard Events) — keydown, keyup
- Touch Events — pekskärm
- Pointer Events — kombination mus och pekskärm

Webbteknik 3

Progression (Progression Events) — load, unload, ...

Formulär (Form Events) — change, focus, blur, submit, ...

Fel som inträffar (Error Event) — error

...

click



drag



submit



enter



leave

~~Attribut i HTML-koden~~

~~<tagg onclick="myFunc()">...</tagg>~~

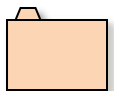
Egenskap till HTML-element i JavaScript

```
elem = document.querySelector("#myElem");  
elem.onclick = myFunc;
```

Med funktionen addEventListener i JavaScript

```
elem = document.querySelector("#myElem");  
elem.addEventListener("click", myFunc);  
...  
...  
elem.removeEventListener("click", myFunc);
```

- + enkelt
 - blandar HTML och JavaScript, utspritt i HTML-koden
-
- + all hantering görs i JavaScript
 - + "renare" HTML-kod
 - man kan endast ha en funktion kopplad till en händelselyssnare
-
- + flera funktioner kan kopplas till samma händelse och HTML-element
 - + man kan enkelt lägga till och ta bort händelselyssnare på elementen



```
element.addEventListener("click", doSomething);
```

```
function doSomething(e) {  
    ...  
    e.egenskap  
    e.metod()  
    ...  
}
```

Allmänt för Event-objektet

Egenskaper

- `currentTarget`
- `target`
- `type`
- ...

Metoder

- `preventDefault()`
- `stopPropagation()`
- ...

För mushändelser

Egenskaper

- `altKey`, `ctrlKey`, `shiftKey`
- `offsetX`, `offsetY`
- `clientX`, `clientY`
- ...

För tangentbords-händelser

Egenskaper

- `altKey`, `ctrlKey`, `shiftKey`
- `code`
- `key`
- ...



Exempel på några händelser för musen

- click
- dblclick
- mouseenter
- mouseleave
- mouseover
- mouseout
- ...

Händelser för tangentbordet

- keydown
- keyup

Testa det som körs i LiveServer på mobilen

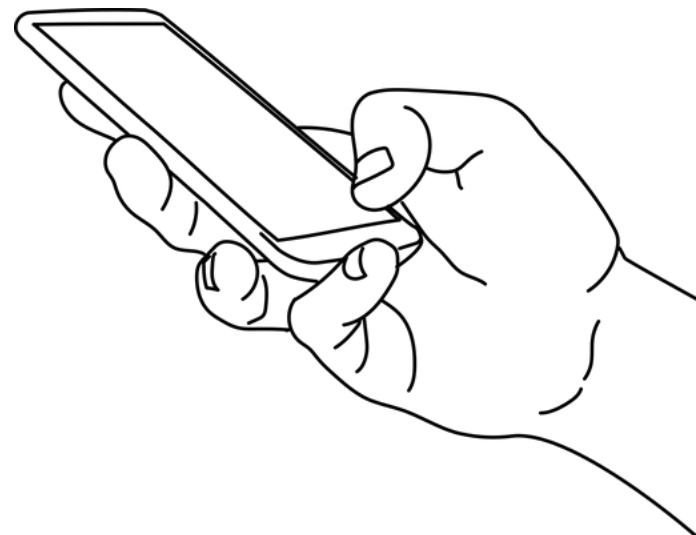
<https://youtu.be/uRYHX4EwYYA?si=KFb0qTy96bHZpLNd>

CSS

:hover

JavaScript

mouseenter och mouseleave

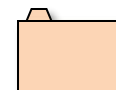


```
@media screen and (hover: hover) {  
    /* Kod för enhet där hover fungerar */  
}
```

```
usingTouchScreen = ("ontouchstart" in window);
```

```
if (usingTouchScreen) {  
    // Kod för pekskärm  
}  
else {  
    // Kod för "vanlig" skärm  
}
```

Händelser för pekskärm tas upp i kursen Webbteknik 3.



F4-ex2

Studera exempel och övningar på js.enur.se

Exempel 2.14 - 2.15 och 3.1 – 3.3

- Math och Array
 - Slumptal för val av värde ur en lista (array)
- Datum och tid
 - Date-objektet
 - metoder för att ta fram delar av datum
 - getTime() för tidsmätning
- Händelser
 - Event-objektet och olika händelser
 - några egenskaper och metoder i event-objektet
 - mushändelser
 - tangentbordshändelser

Slutför uppgift U1b