

PGR102 – Intro til Programmering

Andre repetisjonsforelesning

Jeg legger ut alt materialet fra forelesningene her:

<https://github.com/gulpipose/PGR102-H20-Intro-til-programmering>

Kontakt meg på Discord, Facebook eller: orbgur18@egms.no

Hva skal vi repetere i dag?

If/Else forts.

Funksjoner

`<input>`

Event

Om vi rekker: Oppgaver og litt til?

IF og ELSE - kontrollstrukturer

Fortsetter med spilleksempellet fra forrige uke

Vi vil printe ut på nettsiden hvem det er som har vunnet!

```
var rolandosScore = 10;  
var gurosScore = 20;
```

IF og ELSE - kontrollstrukturer

```
if(rolandosScore > gurosScore) {  
    alert(`Gratulerer! ${rolandosName}!`);  
} else if(rolandosScore < gurosScore){  
    alert(`Gratulerer! ${gurosName}!`);  
} else if(rolandosScore === gurosScore){  
    alert(`Det ble uavgjort :S `);  
}
```

IF og ELSE - kontrollstrukturer

La oss gjøre eksempelet litt mer komplisert

Nå må spilleren ha *minst 50 poeng* for å vinne og *høyest* score for å vinne.

Hvordan skal vi nå få printet ut riktig output til brukeren?

IF og ELSE - kontrollstrukturer

Logiske operasjoner

Lar oss sjekke for mer komplekse tilfeller i en kontrollstruktur.

I dette faget ser vi på: || OR – «ELLER»

&& AND - «OG»

IF og ELSE - kontrollstrukturer

Logiske operatoren OR ||

Minst én av betingelsene må være sanne for at hele uttrykket skal regnes som sant.

1 er mindre enn 10. Dette er TRUE 5 er IKKE mer enn 10. Dette er FALSE

```
if(1 < 10 || 5 > 10); {  
  
    alert(`Dette er sant!`);  
  
}
```

127.0.0.1:50970 says

Dette er sant!

OK

Fordi hvertfall én av de er sanne
blir alerten skrevet ut! 😊

IF og ELSE - kontrollstrukturer

Logiske operatoren AND &&

Begge betingelsene må være sanne for at hele uttrykket skal regnes som sant.

1 er mindre enn 10. Dette er TRUE 15 er mer enn 10. Dette er TRUE

```
if(1 < 10 && 15 > 10); {  
  
    alert(`Dette er sant!`);  
  
}
```

127.0.0.1:50970 says

Dette er sant!

OK

Fordi begge deler er sant blir alerten skrevet ut! 😊

OBS! Hadde én av de vært false hadde IKKE alerten blitt skrevet ut!

IF og ELSE - kontrollstrukturer

Bruk av logiske operatorer – noen eksempler

Vi kan legge til flere på rad for å sjekke mer enn to ting.

```
if(1 < 10 || 15 < 10)
```

//Dette blir true!

```
if(4 < 10 && 8 < 6)
```

//Dette blir false!

```
if(3 > 1 && 9 < 10 && 15 > 10)
```

//Dette blir true!

```
if(7 > 10 || 20 === 10 || 15 < 10)
```

//Dette blir false!

IF og ELSE - kontrollstrukturer

La oss gjøre eksempelet litt mer komplisert

Nå må spilleren ha *minst 50 poeng* for å vinne og *høyest* score for å vinne.

Hvordan skal vi nå få printet ut riktig output til brukeren?

Vi bruker den logiske operatoren &&!

IF og ELSE - kontrollstrukturer

```
if(rolandosScore > 50 && rolandosScore > gurosScore) {  
    alert(`Det ble uavgjort :S`);  
} else if(rolandosScore > rolandoScore){  
    alert(`Det ble uavgjort :S`);  
} else if(rolandosScore > 50 &&  
    rolandosScore === gurosScore){  
    alert(`Det ble uavgjort :S`);  
} else {  
    alert(`Ingen av dere vant :(`);  
}
```

127.0.0.1:50970 says

Ingen av dere vant :(

OK

var rolandosScore

var gurosScore = 4

Funksjoner

Hva er en funksjon?

En blokk med kode.

Kan skrives hvor som helst. Kjøres først når man kaller på den.

Vi kan kalle på den så mange ganger vi vil.

```
var gurosScore = 25;
```

```
function giveGuroPoint() {  
  
    gurosScore++  
  
}
```

```
giveGuroPoint(); //Når vi kaller på funksjonen blir  
gurosScore 26
```

Funksjoner

Hvorfor funksjoner?

Gir oss muligheten til å bruke samme kode om og om igjen.

Vi kan også bestemme *når* koden skal kjøres.

Spesielt nyttig sammen med `<input>` og *event*.

<Input>

En generell html-tag vi kan bruke for brukerinput

Vi setter type=«» for å sette hva slags html-tag den skal se ut som.

Lager en knapp ved bruk av <button>.

```
<button id=«submit-btn»>...</button>
```

Lager en knapp ved bruk av <input>.

```
<input id=«submit-btn» type=«button»  
value=«...» />
```

De blir visuelt helt like på nettsiden, så hva er egentlig forskjellen?

<Input>

Gir oss tilgang til et sett verktøy

Når vi bruker <input> får vi tilgang til å blant annet:

- Hente ut innholdet i et tekst-input felt - **.value**
- Si noe om hva som skal skje når man trykker på en knapp - **.onClick = ...**

Finnes flere også(se Rolandos slides!)

Events

Hver gang du «gjør noe» på maskinen din fyrer dette av et «event»

Et veldig vanlig konsept innen applikasjonsprogrammering.

Vi skal hovedsaklig se på click-eventet – altså hva skjer når du klikker på noe?

Men vit at det finnes fler! 😊