

### 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?



#### Fortsetter med spilleksempelet fra forrige uke

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

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



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



#### 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?



#### **Logiske operatorer**

Lar oss sjekke for mer komplekse tilfeller i en kontrollstruktur.

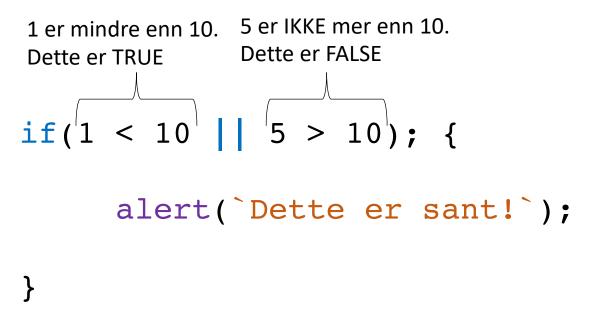
I dette faget ser vi på: OR – «ELLER»

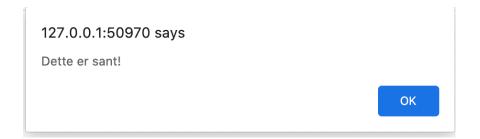
&& AND - «OG»



#### Logiske operatoren OR | |

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



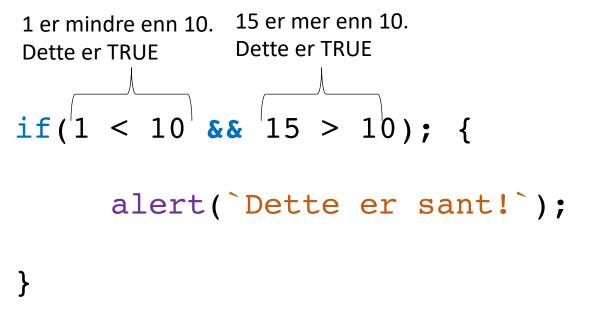


Fordi hvertfall én av de er sanne blir alerten skrevet ut! ©



#### Logiske operatoren AND &&

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



127.0.0.1:50970 says

Dette er sant!

Fordi begge deler er sant blir alerten skrevet ut! ©

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



#### Bruk av logiske operatorer – noen eksempler

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



#### 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(rolandosScore > 50 && rolandosScore > gurosScore) {
                                                                        me}!`);
                       127.0.0.1:50970 says
                                                                        > rolandoScore){
                       Ingen av dere vant :(
var rolandosScore
                                                                        !`);
                                                                OK
var gurosScore = 4
                                                                        core > 50 &&
                                               rolandosScore === gurosScore) {
                                       alert(`Det ble uavgjort :S `);
                               } else {
                                       alert(`Ingen av dere vant :(`);
```



# 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 htmltag den skal se ut som. Lager en knapp ved bruk av <button>.

```
<button id=«submit-btn»>...
```

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! ©