# OPPGAVE: «Gjett et tall»-lek Fra algoritme til kode

«Gjett et tall!»-lek fungerer slik at noen tenker et tall, og du må gjette tallet. For hvert forslag du gir får du tips om ditt tall var for høyt eller for lav.

En algoritme er en sett med instruksjoner som mennesker forstår, som kan brukes av programmerer for å skrive kode. Jobben til programmerer er å forstå hva hver instruksjon gjør og oversette det til python kode.

# Oppgaver

1. **Skrive kode for «Gjett et tall»-lek ved hjelp av algoritmen du ser i den blå boks under:**
2. Import random bibliotek
3. Få maskinen til å velge et tilfeldigtall ved hjelp av random-bibliotek og lagre det i en variabel

(Tips: her trenger du å bruke funksjon randint() og en variable)

1. Ber brukeren (den som kjører program) om skrive tallet hen gjetter er tallet maskin valgte, og lagre det som en heltall i en variable (Tips: her trenger du å bruke int() og input())
2. Så lenge svaret fra bruker ikke er lik tallet maskin valgte gjør følgende: (tips, her trenger du while)
   1. Hvis tallet som bruker skrev er mindre enn tallet til maskinen:
      1. Skriv til bruker at tallet var for lav
   2. Hvis svaret fra bruker er større enn tallet til maskinen:
      1. Skriv til bruker at tallet var for stor
   3. Be bruker om å gi deg et nytt tall og lagre svaret i samme variabel som du brukte i steg 3.
3. Når svaret fra bruker er lik tallet maskinen valgte print «Du klarte det!»
4. **Tell hvor mange forsøk man bruker for å gjette tall**

Bruk en variable for å telle hvor mange ganger bruker har gjettet. Tips: du må oppdatere den inn i while-løkke)

1. **Begrense brukerens forsøk**

Til nå har brukeren hatt uendelig mange forsøk. Skrive om koden din slik at brukeren bare får 12 forsøk? (tips – du trenger en ekstra variable og vilkår i while)

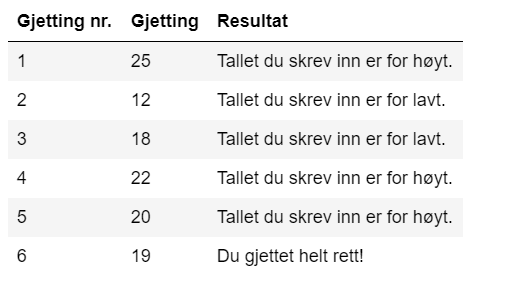
1. **Halveringsmetode - Minimisere antall gjett**

Hvor mange ganger må du gjette før du kommer fram til rett svar? Hvordan kommer vi raskest fram til rett svar? Nedenfor er det listet opp noen framgangsmåter.

* Start på 0, og prøv ett og ett tall oppover til du får det rette.
* Start på 0, og prøv hver hele tier oppover til du kommer for høyt, ta deretter ett og ett tall oppover i det aktuelle tierintervallet.
* Start på 0, og prøv hvert 20-tall oppover til du kommer for høyt, ta deretter ett og ett tall i det aktuelle 20-intervallet.
* Start på 0, og prøv hvert 20-tall oppover til du kommer for høyt, prøv deretter tieren midt inne i 20-intervallet for til slutt å ta ett og ett tall i det rette tierintervallet.

Hvilken av disse tror du er best? Den første vil i alle fall ta lang tid ...

Halveringsmetoden er en systematisk metode å gjette seg fram til rett svar på som vil være raskest i de fleste tilfeller. Dersom den blir brukt i programmet ovenfor, vil det kunne se slik ut når øvre grense for tall er 50:

**

1. Sjekk på hvilken tall blir valgt som gjetting. Kan du finne et mønster?
2. Skriv en algoritme (sett med instruksjoner) for halveringsmetoden som er brukt her:
3. **Maskin skal gjette nå**

Skriv et nytt program hvor det er maskin som gjette på et tall du har tenkt. Programmet skal foreslå først et tall til deg og du skal fortelle om tallet er for høyt eller for lav. Maskin skal bruke halveringsmetoden for å gjette seg fram til tallet du tenkte.

Her er algoritme for valg av tall med halveringsmetoden:

* Gjett på det tallet som er i midten av intervallet tallet skal ligge i.
* Dersom tallet er for høyt, sett tallet som ny øvre grense av intervallet.
* Dersom tallet er for lavt, sett tallet som ny nedre grense av intervallet.
* Dersom tallet er rett, er vi i mål.
* Gjør alt om igjen ifra starten helt til tallet er re

# «Gjett et tall» lek

## Oppvarming

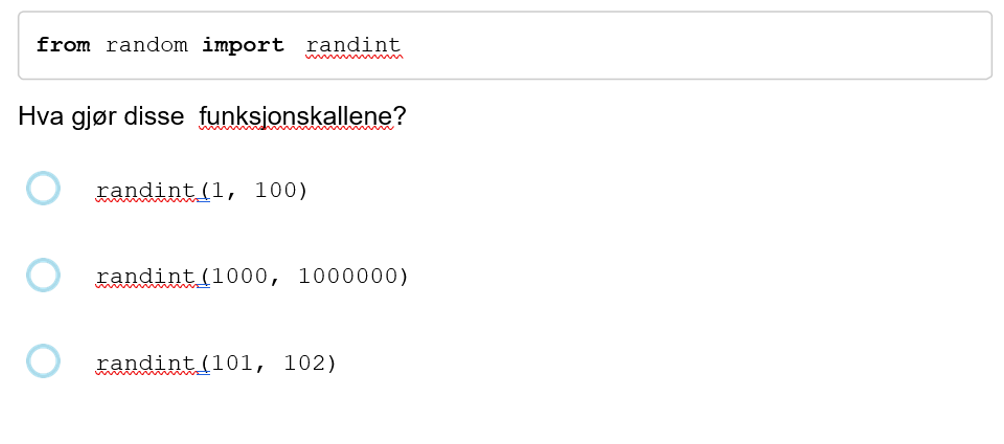
### 1 Hvordan generere en tilfeldig tall

Random er en module med kode som noen andre har skrevet som vi kan bruke. «Random»-module kan brukes for å generere random verdier.

For å bruke en module som andre har skrevt bruker vi komando *import*. Hvis du vil bruke bare en av funksjoner som er med i modulen bruker du *from* module\_navn *import* funksjons\_navn.

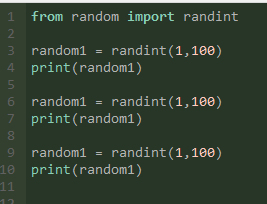
Randint() er en funksjon som generere random heltall (rand int 🡪 random integer).

Test denne i konsoll-delen av Thonny og sjekke hva funksjonskallene gjør?



Skriv og kjør denne koden i en program og kjør koden. Hva gjør denne programmet?

Svar:



### 2. Lese et numerisk svar fra bruker



3. Kontroll strukturer du trenger for å kode «Gjett et tall»-lek