

Raadspel (Guessing Game) Uitdaging

Laten we `while`-loops gebruiken om een raadspel te maken.

De uitdaging:

Schrijf een programma dat een willekeurig geheel getal kiest van 1 tot 100, en laat spelers het getal raden. De regels zijn:

1. Als de gok van een speler kleiner is dan 1 of groter dan 100, zeg dan “OUT OF BOUNDS”
2. Bij de eerste beurt van een speler, als hun gok `.is`
 - binnen 10 van het getal, retourneer “WARM!”
 - verder dan 10 van het nummer, retourneer “COLD!”
3. Bij alle volgende beurten, als een gok is
 - dichterbij het getal dan de vorige gok, retourneer “WARMER!”
 - verder van het nummer dan de vorige gok, retourneer “COLDER!”
4. Wanneer de gok van de speler gelijk is aan het aantal, vertel hem dan dat hij goed heeft geraden *en* hoeveel keer het geraden heeft!

U kunt dit helemaal opnieuw proberen of de onderstaande stappen volgen. Er is een apart Solution-notebook meegeleverd. Succes!

Kies eerst een willekeurig (`random`) geheel getal van 1 tot 100 met behulp van de `random` module en wijs het toe aan een variabele

Opmerking: `random.randint(a,b)` retourneert een willekeurig geheel getal in het range `[a, b]`, **inclusief** beide eindpunten.

Druk vervolgens een inleiding tot het spel af en leg de regels uit

Maak een lijst om gissingen op te slaan Hint: nul is een goede tijdelijke aanduiding. Het is handig omdat het evalueert naar “False”

Schrijf een `while`-lus die om een geldige gok (`guess`) vraagt. Test het een paar keer om er zeker van te zijn dat het werkt.

```
while True:
```

```
    pass
```

Schrijf een `while`-lus die de gok van de speler vergelijkt met ons getal. Als de speler correct raadt, breek dan uit de lus. Vertel de speler anders of het warmer of kouder is en blijf vragen om te raden. Enkele tips: * Het kan helpen om eerst alle mogelijke combinaties op papier te schetsen (sketch)! * U kunt de functie `abs()` gebruiken om het positieve verschil tussen twee getallen te vinden * Als u alle nieuwe gissingen (`guesses`) aan de lijst toevoegt, wordt de vorige gissing gegeven als `guesses[-2]`

```
while True:

    # we can copy the code from above to take an input

    pass
```

Dat is het! Je hebt zojuist je eerste game geprogrammeerd!

In de volgende sectie zullen we leren hoe we sommige van deze repetitieve acties kunnen omzetten in *functies* die kunnen worden aangeroepen wanneer we ze nodig hebben.

Goed gedaan!