

If/elif/else

If/elif/else

Hingað til höfum við verið að skrifa forrit sem fylgja einföldu flæði, þ.e. við tökum inn inntak og skilum út úttaki eftir útreikningar skref. Hvað ef við viljum hafa meiri sveigjanleg forrit sem breyta hegðun eftir inntaki? Sjáum dæmi:

```
import math
radius = int(input("Please input radius: "))
area = math.pi * (r ** 2) # formula for the area of a circle
print("The area of the circle with radius", radius, "is:", area)
```

Hér að ofan er vandamál ef notandinn ákveður að slá inn neikvæða tölu, þar sem hringur getur ekki haft neikvæðan radíus. Lausnin er if og else!

```
import math
radius = int(input("Please input radius: "))
area = math.pi * (radius ** 2) # calculate the area of a circle
if radius >= 0:
    print("The area of the circle with radius", radius, "is:", area)
else:
    print("The radius can not be a negative number!")
```

Boolean



Boolean gagnatýpan getur aðeins verið eitt af tveimur gildum, annað hvort **True** eða **False**. Notað gjarnan við stjórna flæði á forriti, hvað á að keyra og hverju á að sleppa

Aðeins tvö gildi: **True, False**

Mikilvægir samanburðarvirkjar/operators

Sama sem	==	$3 == 3 \rightarrow \text{True}$	$3 == 2 \rightarrow \text{False}$
Ekki Sama sem	!=	$3 != 2 \rightarrow \text{True}$	$3 != 3 \rightarrow \text{False}$
Stærra en	>	$3 > 2 \rightarrow \text{True}$	$2 > 3 \rightarrow \text{False}$
Minna en	<	$2 < 3 \rightarrow \text{True}$	$3 < 2 \rightarrow \text{False}$
Stærra eða jafnt og	>=	$3 >= 3 \rightarrow \text{True}$	$2 >= 3 \rightarrow \text{False}$
Minna eða jafnt og	<=	$2 <= 3 \rightarrow \text{True}$	$4 <= 3 \rightarrow \text{False}$

If

```
boolean_value = True
if boolean_value:
    print("The boolean_value is True")
```

Við skulum athuga if setninguna fyrir ofan og greina hvernig hún virkar:

- **If *expression*** – til að búa til if setningu, þarf að skrifa if og síðan yrðingu. Kóði keyrir einungis ef yrðingin er sönn. Hér að ofan er Boolean_value breytan True og þar af leiðandi mun if setningin keyra.
- **Block(bálgur)** – Allur kóði sem við viljum keyra ef if yrðingin er sönn. Hver lína þarf að vera með 4 space bil (eða einn Tab takka (dálkahak)) á eftir yrðingunni svo að forritið átti sig á hvað á að keyra. Fyrir ofan mun skel birta „The Boolean_value is True“. Ef if setningin keyrir í gang

```
if boolean_value:
    print("This is inside of the if block") # Notice the 4 spaces before
    print("This will only print if boolean_value is True")
print("This is outside the if block")
print("This will print regardless if the boolean_value is True or False")
```

Samanburðarvirkjar == og !=

Hingað til höfum við aðeins unnið með virkja sem reikna út samlagningu/frádrátt/margföldun heiltalna og rauntalna. Stundum er vilji til þess að athuga hvort að eitthvað ákveðið skilyrði er náð

== - Þetta er **sama sem** merkið. Athugið að það eru tvö samasemmerki notuð til að útskurða samanburð. Merkið er notað til að athuga hvort vinstri hlið samasemmerkisins er sú sama og hægri hlið merkisins

!= - Þetta er **Ekki sama sem** merkið. Notað til að athuga hvort að báðar hliðar yrðingarinnar séu ekki það sama

```
number = int(input("Enter a number: "))  
# Runs only if number is equal to 3  
if number == 3:  
    print("you typed in the number 3")
```

```
number = int(input("Enter a number: "))  
# Runs only if number is NOT equal to 3  
if number != 3:  
    print("you typed in any other number than 3")
```

Samanburðarvirkjar > og <

Einnig er hægt að athuga hvort ein tala sé stærri eða minni en hin. Stundum viljum við aðeins vinna með tölur sem eru stærri en 0 eða aðeins vinna með tölur sem eru á bilinu 1 til 10.

> - Þetta er **Stærri en** merkið. Það athugar hvort að vinstri hlið yrðingarinnar sé stærri en hægri hlið yrðingarinnar. T.d. er $3 > 1$ True.

< - Þetta er **Minna en** merkið. Það athugar hvort að vinstri hlið yrðingarinnar sé minni en hægri hlið yrðingarinnar. T.d. er $3 < 1$ False.

Ein þumalputtaregla er að goggurinn (< eða >) er alltaf að „benda“ á töluna sem á að vera minni.

```
number = int(input("Enter a number: "))  
# Runs only if number is more/higher than 3  
if number > 3:  
    print("your number is more than 3")
```

```
number = int(input("Enter a number: "))  
# Runs only if number is less/lower than 3  
if number < 3:  
    print("your number is less than 3")
```

> og < - dæmi

Athugum nokkur áhugaverð dæmi

```
print("4 > 3.0 is", 4 > 3.0)
print("2 < 2.0 is", 2 < 2.0)
print('"a" < "b"', "a" < "b")
print('4 > "3" is', 4 > "3")
```

```
4 > 3.0 is True
```

```
2 < 2.0 is False
```

```
"a" < "b" True
```

```
TypeError: '>' not supported between instances of 'int' and 'str'
```

- Gogga samanburður virkar á milli heiltalna og rauntalna
- Hægt er að bera saman tvo strengi (**Förum betur yfir hvernig það virkar síðar**)
- Ekki er hægt að bera streng saman við int/float

Samanburðarvirkjar \geq og \leq

Þessir virkjar eru samanbland af $<$ eða $>$ og $==$. Athuga hvort talan vinstra megin sé stærri/minni eða jafnt og talan hægra megin við virkjann.

\geq - Þetta er **Stærri en eða jafnt og** merkið.

\leq - Þetta er **Minna en eða jafnt og** merkið.

```
number = int(input("Enter a number: "))  
# Runs only if number is more/higher than 3 or equal to 3  
if number  $\geq$  3:  
    print("your number is more than 3 or it is equal to 3")
```

```
number = int(input("Enter a number: "))  
# Runs only if number is less/lower than 3 or equal to 3  
if number  $\leq$  3:  
    print("your number is less than 3 or it is equal to 3")
```


Else

```
number = int(input("Enter a number"))
if number >= 3:
    print("The number is higher or equal to 3")
else:
    print("The number is less than 3")
```

Ef við viljum að forritið hegði sér öðruvísi ef if yrðing gengur ekki upp, notum við else.

Else setning þarf if setningu fyrir framan sig, öll tilvik sem if setningin grípur ekki tekur else að sér. Eins og dæmið fyrir ofan, ef talan er lægri en 3, þá mun skel birta „The number is less than 3“.

Eins og með if, þarf 4 bil (eða einn Tab takka (dálkahak)) til að tilgreina hvar else blokkinn byrjar og endar.

```
if boolean_value:
    print("This is inside the if block") # Runs if boolean_value is True
else:
    print("This is inside the else block") # Runs if boolean_value is False
    print("This is also inside the else block!")
print("This is neither in the if block nor the else block")
```

Elif

Skoðum fyrst eitt dæmi af forriti sem prentar út hversu marga tölustafi tala inniheldur:

```
number = int(input("Enter a number(Higher than 0): "))
if number < 10:
    print("The number has 1 digit")
if number < 100:
    print("The number has 2 digits")
if number < 1000:
    print("The number has 3 digits")
else:
    print("The number has more than 3 digits")
```

Ef við gefum inn inntakið 54000, fáum við:

```
The number has more than 4 digits
```

Aftur á móti ef við sláum inn 7 sem inntak fáum við:

```
The number has 1 digit
The number has 2 digits
The number has 3 digits
```

Elif – áframh.

Við sjáum að í síðustu glæru prentar hann út 3 línur, þegar við viljum aðeins eina. Forritið les hverja if setningu sér og else setningin á bara við seinustu if setninguna. Til að komast í kringum þetta er hægt að nota **elif** sem sendur fyrir **else if**

```
number = int(input("Enter a number(Higher than 0): "))
if number < 10:
    print("The number has 1 digit")
elif number < 100:
    print("The number has 2 digits")
elif number < 1000:
    print("The number has 3 digits")
else:
    print("The number has more than 4 digits")
```

Ef slegið er inn 7 sem inntak, fáum við:

```
The number has 1 digit
```

Elif – áframh.

Elif – elif setning þarf að vera með yrðingu. Hún keyrir einungins ef if/elif setningin á undan henni er False og ef yrðingin innan við elif setningunnar er True.

Eins og if og else, þá þarf 4 bil eða 1 tab til að tilgreina hvar blokkinn byrjar að keyra.

Uppröðun if, elif og else skiptir miklu máli þar sem Python les fyrst if setninguna, síðan elif setninguna og síðan else

```
if my_bank_balance >= your_bank_balance:
    print("I'm richer than you")
elif my_bank_balance == your_bank_balance:
    print("We are equal") # This line will never run
else:
    print("Damn, you are richer than me")
```

Athugið að í dæminu fyrir ofan mun elif-blokkinn aldrei keyra. If setningin er nú þegar að athuga hvort my_bank_balance er jafnt of your_bank_balance. Ef if yrðingin er sönn, mun forritið hunsa elif-blokkina og else-blokkina

and og or

Stundum nægir ekki að hafa eina yrðingu innan í if/elif setningu. Við notum þá **and** og **or** til að keðja saman yrðingar

x **and** y – yrðingin er aðeins True ef x **og** y eru bæði True

x **or** y – yrðingin er aðeins True ef x **eða** y eru True

Dæmi um **and**:

```
my_secret_password = "password123"
user_password = input("Please enter the password: ")
user_password_retry = input("Please re-enter the same password: ")
if user_password == my_secret_password and user_password == user_password_retry:
    print("You know the password, you can enter")
else:
    print("You don't know the password, or you retyped it wrong, you may not enter")
```

Hér að ofan mun forritið aðeins hleypa þér í gegn ef að þú slærð inn “password123” og slærð síðan aftur inn “password123”. If yrðingin þarf að vera True and True til að keyra if-blokkina.

and og or – áframh.

Dæmi um *or*:

```
age = int(input("Enter your age: "))  
if age < 16 or age > 65:  
    print("You don't have to pay to enter")  
else:  
    print("You have to pay 500kr to enter")
```

Hér að ofan mun if setningin keyra ef aldurinn þinn er 15 ára eða yngri EÐA 66 ára eða eldri.

and og or – áframh.

Það er ákveðin forgangsröð sem gildir fyrir **and** og **or**. **And** hefur ávallt meiri forgang en **or**.

```
if 5 > 6 or 10 == 10 and 4 != 4:  
    print("True")  
else:  
    print("False")
```

False

Hér er prentað út false því að forritið athugar fyrst **and**. Föllum yfir útreikningana skref fyrir skref:

	<code>5 > 6 or 10 == 10 and 4 != 4</code>	(Fyrst er athugað yrðingar hjá and)
→	<code>5 > 6 or True and False</code>	(True and False skilar False)
→	<code>5 > 6 or False</code>	(Næst er reiknað út yrðingar hjá or)
→	<code>False or False</code>	(False or False skilar False)
→	<code>False</code>	

and og or – áframh.

Hægt er að breyta forgangsröðinni með svigum, svipað og er gert í hefðbundinni stærðfræði:

```
if (6 > 5 or 10 != 10) and 4 == 4:  
    print("True")  
else:  
    print("False")
```

True

Hér er prentað út True því að forritið athugar fyrst **or** vegna sviga. Förum yfir útreikningana skref fyrir skref:

→ (6 > 5 **or** 10 != 10) **and** 4 == 4
→ (True **or** False) **and** 4 == 4
→ True **and** 4 == 4
→ True **and** True
→ True

(Fyrst er athugað yrðingar hjá **or** vegna sviga)
(True **or** False skilar True)
(Næst er reiknað út yrðingar hjá **and**)
(True **and** True skilar True)

Setningavirkjar (comparison operators)

Ein afleiðing úr rökfræði sem hefur mikla þýðingu í forritun er notkun setningavirkja. Þau eru líka kölluð setningatengi en þau tengja saman föll og breytur innan skipana. Þið kannist við flest þeirra en dæmi um setningavirkja sem við notum eru: **and**, **or**, **xor**, **not**

Til að skilja hvernig setningavirkjarnir virka notum við sanntöflur sem sýna okkur hvert úttakið verður eftir því hvort breytturnar sitthvorumegin við setningatengið eru **True** eða **False**.

Sanntöflur fyrir *or* og *and*

Skilgreiningarsanntafla eða-
tengingar

p	q	$p \vee q$
s	s	s
s	ó	s
ó	s	s
ó	ó	ó

Skilgreiningarsanntafla fyrir
og-tengingu

p	q	$p \wedge q$
s	s	s
s	ó	ó
ó	s	ó
ó	ó	ó

Hvernig á að nota if með input

Þegar unnið er með tölur sem koma frá input þarf sérstaklega að passa upp á að inntakið er breytt í rétt gagnatag áður en unnið er með tölurnar. Python mun ekki henda inn villu ef þú berð saman tvo strengi sem innihalda tölu, þannig að óvænt villa gæti komið upp án villuskilaboða!

```
first_number_str = input("Enter number 1: ")
second_number_str = input("Enter number 2:")

first_number_int = int(first_number_str)
second_number_int = int(second_number_str)

if first_number_int > second_number_int:
    print(first_number_int, "is larger than", second_number_int)
elif first_number_int < second_number_int:
    print(second_number_int, "is larger than", first_number_int)
else:
    print("The numbers are equal")
```

Það er góður siður í Python að enda breytu nafn með „_str“ ef breytan er strengur, „_int“ fyrir int, „_float“ fyrir float og „_bool“ fyrir boolean. Þetta er gott þegar þú veist að þú þarft að skipta um gagnatag eins og dæminu fyrir ofan.

$a \% b == c$

Modulo getur reynst hentugur í alls konar tilfellum. Eitt þeirra tilvika er þegar verið er að athuga hvort tala sé oddatala eða slétt.

Ef við deilum slétta tölu með 2 mun hún alltaf fullkomlega deila með 2.

Ef við deilum oddatölu með 2 mun hún alltaf skila 1 sem afgang.

T.d. $3 \% 2$ verður 1 og $10 \% 2$ verður 0

```
number = int(input("Enter a number: "))
if number % 2 == 1:
    print("The number is odd")
elif number % 2 == 0:
    print("The number is even")
```

(Athugið að í staðinn fyrir **elif** væri hægt að nota **else**, **elif** notað hér til að sýna eiginleika modulo)

Ef að $a \% b$ er 0, vitum við að **a** er deilanlegur með **b**. Ef það er enginn afgangur frá modulo, þýðir það að talan deilir fullkomlega með hinni tölunni

```
number = int(input("Enter a number: "))
if number % 5 == 0:
    print("The number", number, "is divisible by 5")
```

Innri if lykkjur

Hægt er að hafa if lykkju innan í if lykkju. Þetta getur verið hentugt ef þú vilt athuga tvö skilyrði séu uppfyllt, en stundum er betra að nota **and** eða **or**.

```
age = int(input("Enter your age: "))
height = int(input("Enter your height: "))
if age > 13:
    print("You are old enough to ride this rollercoaster")
    if height > 150:
        print("and you are tall enough to ride this rollercoaster")
    else:
        print("but you are not tall enough to ride this rollercoaster")
else:
    print("You are not old enough to ride this rollercoaster")
```

```
Enter your age: 15
Enter your height: 140
You are old enough to ride this rollercoaster
but you are not tall enough to ride this rollercoaster
```

Margar leiðir til að skirfa sama virkni

Stundum skiptir máli að velja hvernig er nálgast dæmi í forritun.

Dæmi: Skrifum forrit sem tekur inn tölu og lætur notenda vita ef talan er stærri en 55 eða minni eða jafnt og 55 og hvort að talan sé deilanleg með 3. Hér er leið númer 2 mun styttri, jafnvel þótt þær geri það sama

```
if number > 55 and number % 3 == 0:
    print("The number is larger than 55")
    print("The number is divisible by 3")
elif number <= 55 and number % 3 == 0:
    print("The number is lower than or equal to 55")
    print("The number is divisible by 3")
else:
    if number > 55:
        print("The number is larger than 55")
    else:
        print("The number is lower than or equal to 55")
```

Leið 1

```
if number > 55:
    print("The number is larger than 55")
else:
    print("The number is lower than or equal to 55")
if number % 3 == 0:
    print("The number is divisible by 3")
```

Leið 2

Algeng villuskilaboð

Hér eru nokkrar lúmskar villur sem þið gætuð lent í

Syntax Error – Þegar vitlaust samanburðarmerki er notað

```
SyntaxError: invalid syntax
```

Mjög algengt fyrir byrjendur að nota aðeins eitt samasemmerki (=) við samanburð, muna að nota tvö

```
if 3 = 3: # Should be 3 == 3
    print("True")
```

TypeError – Villa með samanburð strengs og heiltölu

```
TypeError: '>' not supported between instances of 'int' and 'str'
```

Kemur upp þegar reynt er að bera saman streng og heiltölu með goggi

```
number = input("Enter a number: ") # Input returns a string!
if number > 3: # str > int !!!
    print("Your number is larger than 3")
```

Fleiri algeng villuskilaboð

Syntax Error – Gleymir yrðingu í *if* eða *elif*

```
SyntaxError: invalid syntax
```

Kemur upp þegar gleymt er að setja yrðingu innan í *if* eða *elif*, terminal segir til á hvaða línu yrðingu vantar

```
number = int(input("Enter a number: "))  
if 0 < number < 3:  
    print("your number is either 1 or 2")  
elif: # No expression here!  
    print("Your number is either 3 or 4")  
else:  
    print("Your number is not 1,2,3 or 4")
```