

Kynning á JupyterLab og Python

Ef annað er ekki tekið fram á að byrja á að keyra reitina í þessari bók (smella á þá og slá á control/enter) og fylgja svo leiðbeiningum. Ef þær vantar getur verið sniðugt að gera einhverjar breytingar að eigin vali og skoða áhrifin.

Útprentun

- Jupyter skrifar sjálfkrafa niðurstöðu segðar sem er sett aftast í reit
- Nota má `print`-fallið til að skrifa út niðurstöður frammar í reitnum.

1. Prófið að setja `;` á eftir y í öftustu línunni.

```
In [ ]: x = 1 + 2          # Myllumerki (tvíkross fyrir komment)
        print(x, 2 + 2)

        y = x + 2
        x + 1, y          # Til að sýna tvær niðurstöður má setja kommu á milli
```

Stutt æfing í Jupyter-ritvinnslu

Skoðið ósniðna textann í þessum reit með því að tvísmella á hann. Sníðið svo textann með control/enter.

- Prófið að búa til reit fyrir ofan þennan með `esc-a` og fyrir neðan með `esc-b` (above/below). Breytið neðri reitnum í textareit með `esc-m` (eða smellið á Markdown) og aftur í forritsreit með `esc-y` (eða smellið á Code). Takið eftir að það er `[nn]:` framan við forritsreiti.
- Látið reit vera Markdown og setjið inn fyrirsögn með `##` fremst, og prófið feitletrun og skáletrun með `**feitt**` og `*ská*`. Prófið að afturkalla (*undo*) með control/z (command/z á Mac) og endurtaka (*redo*) með control/shift/z (command/shift/z á Mac). Prófið að eyða þessum reitum með `esc-d-d` og afturkalla eyðinguna með `esc-z` (*cell-undo*).
- Skoðið hér á eftir hvernig settur er forritsbútur inn í textareit (með inndrætti um 4 bil) og prófið að endurtaka leikinn sjálf.

Forrit:

```
x = 3
y = x + 2
print(y)
```

Æfing í JupyterLab-skrám

- Felið skráalistann og sýnið aftur með því að smella á möppuna efst t.v. (neðan við Jupyter-táknið).
- Búið til nýja vinnubók með File–New–Notebook (og veljið svo þessa jafnskjótt aftur). Takið eftir að nýja bókin birtist sem *Untitledxx* í skráalista vinstra megin.
- Skiptið um nafn á nýju bókinni með því að hægri-smella á hana og velja *Rename*.

4. Setjið eitthvað einfalt forrit inn í nýju bókina. Prófið svo að draga flipann með vinnubókarnafninu til ("drag-and-drop") og sýna þannig báðar vinnubækurnar hlið við hlið, hvora ofan við aðra, og aftur eins og byrjað var með, eina í einu.
5. Eyðið nýju bókinni með hægri smelli og "Delete".

Breytur og tög

Breytur hafa ekki tög (type) eins og í Java, heldur hafa gildi tög. Ekki þarf að skilgreina breytur. Grunttögin eru m.a.:

- rökgildi (bool): `True/False`
- strengir (string) `'strengur', '42', "strengur í tvöföldum gæsalöppum"`
- heiltölur (int): `42`
- kommutölur (float): `-3.14159, 1e100`

```
In [ ]: k = 3
print(type(k))      # til að finna tag gildis má nota type
display(type(k))    # display skrifar öðruvísi en print
k = 3.4
display(type(k))
type('strengur'), type(k > 4), type(False) # þetta skrifast í raun með display
```

Meira um print

```
In [ ]: s = "Strengur"
print(s + s)        # skeytir saman strengjum, ekkert bil
print(s, s)         # setur bil á milli
print(s, s, sep=", ") # separator
print("s = ", end="") # fer ekki í næstu línu
print(s)
```

F-strengir og ný lína (linefeed)

1. Prófið að prenta s með f-streng `f"{s:5}"`
2. Prentið heiltölu með `{k:5}`, `{k:<5}` og `{k:>5}`

```
In [ ]: e = 2.7182818
k = 12
print(f"Jón er {k+1} ára") # slaufusvigar utan um segð
print(f'e = {e:.3f}')      # 3 aukastafir
print(f'e = {e:6.3f}')     # 6 stafa breitt svið, 3 aukast.
s = 'abc'
print(f'{s:<5}\n{s:>5}')    # vinstri jafna, ný lína, hægri jafna í 5 stafa brei
print(f'{s:5}')
```

Strengir

Við getum skeytt saman strengjum með `+` og endurtekið strengi með `*`

1. Búið fyrst til strengi með fornafni ykkar, millinafni (ef við á) og eftirnafni, og búið svo til streng með fullu nafni með `+` (notið `" "` til að fá bilin á milli).
2. Búið til f-streng með `e` með þremur aukastöfum, `"e = 2.718"`

```
In [ ]: a = 'a'
b = 'xy'
```

```
x = 1
e = 2.71828
print(len(b))           # lengd strengsins
print(a + b)            # samskeyting
print(a*5 + b*3)        # margföldun er eins og margir plúsar
print("x = " + str(x))  # str breytir heiltölu í streng
print("e = " + str(e))  # og líka kommutölu – en ekki hægt að stjórna aukastafa
print("x = " + x)       # tölum er ekki breytt sjálfkrafa í strengi
```

Hlutstrengir

Við getum vísað í hluta af streng með

- einni tölu, byrjað að telja í 0, t.d. `s[0]`
- bili, `s[0:5]`
- frá byrjun: `s[:5]`, eða út í enda: `s[1:]`
- og talið aftur á bak `s[-1]`

Í reitnum á eftir er fyrsti stafur strengsins skrifaður. Prófið alla hina möguleikana. Prófið líka að velja frá og með sæti 2 til og með næstsíðasta sætisins.

```
In [ ]: s = "abcdefg"
        print(s[0])
```

Útreikningur

Heiltölur hegða sér eins og í Java, nema að þær hafa engin stærðartakmörk. Venjulegar reiknireglur gilda um `+`, `-`, `*`, `/`, `**` er veldi `%` er afgangur úr deilingu og `//` er heiltöludeiling. Nota má gildisgjöf `+=` eins og í Java, en `++` gengur ekki

```
In [ ]: a = 4
        a += 1      # a verður 5
        print(a/3)
        print(a//3)
        print(a % 3)
        print(4**3)
        print(4.0**3)
        print(2**256)
```

Rökgildi (*boolean*)

Notum `==`, `<=`, `>=`, `<`, `>`, `!=` til að bera saman gildi.

Svo má tengja með `and`, `or` og `not` (en ekki eins og í Java, `&&`, `||` og `!`)

```
In [ ]: x = 3
        print(0 <= x and x <= 10)
        print(0 <= x <= 10)      # sama eins og að ofan, skv. stærðfræðihefð
        print(False or x == 4)  # jafngildir "False or (x == 4)"
```

Ef-setningar

Ef-setning framkvæmir blokk skipana ef skilyrði er uppfyllt og aðra blokk ef ekki. Python notar ekki slaufusviga til að afmarka blokkina heldur inndreginn kóði. Oft eru notuð 4 bil og það gerir Jupyter sjálfkrafa.

Það má sleppa **else**-hluta, og það má setja skipun á sömu línu og **if**. Fyrir *else if* er notað **elif**.

1. Prófið að láta x vera 0 og keyrið aftur.
2. Prófið að láta x vera 14 og keyrið aftur.

```
In [ ]: x = 0
if x == 0:
    print("x er núll")
else:
    print("x er ekki núll")
    print(f"x er {x}")
print("Hér er ef-setningin búin") # ekki hluti af if
if x == 10: print("OK")

if 0 <= x < 12:
    print("Fyrir hádegi")
elif 12 < x <= 24:
    print("Eftir hádegi")
elif x == 12:
    print("Hádegi")
else:
    print("Ekki tími")
```

While-lykkjur

Endurtekur skipun meðan skilyrði er satt, skilyrðið er athugað í byrjun eins og í Java.

1. Prófið að láta x fara upp í 50 og skruna (scrolla) úttakinu. Prófið svo að velja *File—Print* og takið eftir að allt úttakið kemur með í útprentun (*print-preview*-glugganum).

```
In [ ]: x = 0
while x <= 50:
    print(x)
    x += 1
```

For-lykkjur

Ólíkt Java er farið í gegn um mismunandi gildi á stýribreytu með því að kalla á sérstakt fall, **range** -fallið.

```
In [ ]: for x in range(6): # 0 til 5, 6 er ekki með
        print(x)
```

```
In [ ]: for x in range(2,6): print(x) # 2 til 5, skipun má vera á sömu línu
for x in range(0,6,2): print(x) # hoppað um 2 og endað í 4
```

Út úr lykkju / næsta umferð

break og **continue** hegða sér eins og í Java.

1. Búið til for-lykkju sem leitar að fyrstu heiltölunni sem uppfyllir jöfnuna $x^2 + 255 = 32x$ með því að lykkja með x frá 0 til 100 og brjótast út úr lykkjunni með **break** ef jafnan er uppfyllt.
2. Útfærið eftirfarandi reiknirit í Python:

```
Fyrir k=1,...,20:
  y := mod(x,5)    (afgangur úr deilingu með 5)
  ef y=0, þá fara í næstu umferð
  prenta x/y
```

Innbyggð föll

Hér framar hafa innbyggðu föllin `type`, `print` og `len` verið kynnt. Meðal annarra innbyggðra falla eru:

<code>abs(x)</code>	Tölugildi <code>x</code>
<code>max(x,y)</code>	Stærri gildið
<code>min(x,y)</code>	Minna gildið
<code>int(x)</code>	Kommutala \rightarrow heiltala (aukastafir skornir af)
<code>float(x)</code>	Heiltala \rightarrow kommutala

Hér á eftir eru fyrstu tvö föllin prófuð. Prófið hin:

```
In [ ]: abs(-3), max(2,7)
```

Stærðfræðiföll

Til að nota stærðfræðiföll þarf að "flytja inn" svonefnda math-einingu (*-module*). Svo þarf að auðkenna föllin með `math.`, t.d. fæst kvaðratrót með `math.sqrt`. Önnur föll eru m.a. `sin`, `cos`, `tan`, `exp`, `log10`, `log` (hornaföllin miðast við radíana og síðastnefnda fallið gefur náttúrulegan logra, \ln) og fastar eru m.a. `pi` og `e`.

1. Reiknið líka $\log_{10}(1000)$
2. og $\sin(\pi/6)$
3. og að lokum $\ln(e^3)$

```
In [ ]: import math
print(math.sqrt(9))
print(math.pi)
```

Föll skrifuð af notanda

Notum `def` til að skilgreina fall, viðföngin hafa engin tög.

```
In [ ]: def f(x):
        fx = 2*x
        return fx

f(2), f('Halló')
```

Lokaæfingin

Búið til fall sem reiknar $f(x) = \sqrt{x^2 + 3}$. Búið svo til forrit sem skrifar út töflu yfir x og $f(x)$ fyrir $x=0,1,\dots,12$ með eftirfarandi sniði (notið m.a. `f`-strengi):

x	$f(x)$
-----	--------

0	1.7321
1	2.0000
...	
12	12.1244

In []: