

COMHAD FÍRICÍ: TEICNEOLAÍOCHT DHIGITEACH GCSE

Aonad 4 COINCHEAPA DON CHEAPADÓIREACTH DHIGITEACH



Teicnící Simplí Láimhseáil Earráidí

Torthaí Foghlama

Ba chóir go mbeadh daltaí ábalta tuiscint a léiriú ar na rudaí seo a leanas agus iad a úsáid:

- bailíochtú sonraí, lena n-áirítear seiceálacha láithreachta, faid, cineáil agus formáide;
- teicnící aimsithe agus ceartaithe maidir le hearráidí comhréire, reatha agus loighce; agus
- teicnící simplí gaisteoireachta earráidí.

Inneachar

- Bailíochtú Sonraí
 - Seiceáil Láithreachta
 - Seiceáil Cineáil
 - Seiceáil Faid
 - Seiceáil Formáide
- Earráidí Ríomhchlárúcháin
 - Earráidí comhréire
 - Earráidí Rite
 - Earráidí loighce
- Craic le Dífhabhtú!
- Láimhseáil Eisceachtaí

Bailíochtú Sonraí

Tá seanrá sa ríomhaireacht: “*Garbage in garbage out.*” – GIGO. Tá sé seo ag tagairt don fhíric nach dtig le ríomhchóras torthaí cearta a tháirgeadh ó shonraí atá mícheart. Tá sé seo – nó ba chóir go mbeadh sé seo – soiléir do dhuine ar bith a thuigeann ríomhairí. Agus sin ráite, ní bhíonn sé seo soiléir i gcónaí don phobal i gcóitinne.

Go ginearálta, ní féidir le ríomhchóras a chinneadh cé acu atá na sonraí a tugadh dó *ceart* nó nach bhfuil. Agus sin ráite, is féidir leis a chinneadh, i roinnt tosca, cé acu atá na sonraí *reasúnta* nó nach bhfuil, nuair a ghlahtar leis cad é an chiall a bhí i gceist. Is féidir an próiseas seo a uathoibriú agus tugtar *bailíochtú sonraí* air.

Bailíochtú Sonraí

“[Data validation] is the automatic checking of data entered into a computer system. Validation involves using the properties of the data to identify any inputs that are obviously wrong.

Validation only proves that the data entered is a reasonable value for the computer to accept. It cannot prove that the data entered is the actual value the user intended.”

BCS Glossary of Computing, 13th edition, p 75.

Mar shampla, má chuireann úsáideoir an luach –5 isteach nuair a iarrtar air a aois a chur isteach, is léir gur earráid é seo (ní féidir le haois a bheith diúltach), agus ba chóir go mbeadh seiceáil uathoibríthe ábalta é a aithint. Ar an dul chéanna, is léir go bhfuil 33 April 2017 mícheart má bhíodhas ag dúil le dáta – arís eile, ba chóir go n-aithneodh seiceáil uathoibríthe seo. Bíonn muid ag tagairt dóibh seo mar *invalid values* agus ba chóir go mbeadh seiceálacha bailíochtaithe ag ríomhchórais nach ligeann do luachanna neamhbhailí a bheith curtha isteach.

Ar ndóigh, ní féidir le seiceáil bailíochtaithe gach earráid a cheapadh. Mar shampla, b'fhéidir go gcuirfeadh úsáideoir an luach 21 isteach mar aois dó, cé go bhfuil sé 22 i bhfírinne. Go ginearálta, ní féidir le seiceálacha bailíochtaithe earráidí den chineál seo a cheapadh.

Baintear úsáid go coitianta as na cineálacha seo de sheiceálacha bailíochtaithe: seiceálacha láithreachta, seiceálacha faid, seiceálacha cineáil agus seiceálacha formáide.

Seiceálacha Láithreachta

Agus sonraí á gcur isteach, b'fhéidir go mbeadh sé inghlactha in amanna roinnt luachanna a fhágáil amach. Mar shampla, agus sonraí teagmhála úsáideoirí á mbailiú, b'fhéidir go mbeadh sé inghlactha don úsáideoir an réimse uimhir theileafóin a fhágáil folamh. Ar an láimh eile, b'fhéidir go mbeadh réimsí eile éigeantach – mar shampla, is beag is fiú sonraí teagmhála a bhailiú gan ainm an úsáideora. Is seiceáil bailíochtaithe í *seiceáil láithreachta* a chinntíonn gur cuireadh sonraí isteach i ngach ceann de na *réimsí éigeantacha*.

Seiceálacha Faid

Is féidir go minic faid chúí a thuar do luachanna áirithe téacs. Mar shampla, b'fhéidir go mbeadh sé réasúnta a rá go mbeadh ar ainm úsáideora bheith níos mó ná 1 caractar amháin agus níos lú ná (abair) 20. Is seiceáil bailíochtaithe é *seiceáil raoin* a chinntíonn go bhfuil luachanna sonraí d'fhad chúí.

Seiceálacha Cineáil

Bíonn ar luachanna sonraí bheith de chineálacha sainiúla de ghnáth. Mar shampla, ní mór d'aois úsáideora a bheith uimhriúil, agus ní mór gur téacs a ainm. Is seiceáil bailíochtaithe í *seiceáil cineáil* a chinntíonn go bhfuil luachanna sonraí den *chineál* cheart.

Seiceálacha Formáide

In amanna, ní mór sonraí a chur isteach i bpatrún nó i bhformáid réamhshainithe. Mar shampla, is minic a chuirtear dátaí isteach mar thrí phéire digítí amhail 27/05/17. Is seiceáil bailíochtaithe í *seiceáil formáide* a chinntíonn go bhfuil luachanna sonraí san *fhormáid* a bhfuiltear ag dúil léi.

Earráidí Ríomhchláir

In amanna, tugtar *fabhtanna* ar earráidí i ríomhchlár. Aon duine a d'fhorbair níos mó ná ríomhchlár den chineál is simplí, beidh a fhios acu nach féidir earráidí a sheachaint agus gur cuid thábhachtach den tasc bogearraí a fhorbairt é fabhtanna a aimsiú agus a dheisiú.

“If debugging is the process of removing software bugs, then programming must be the process of putting them in.”
~Edsger Dijkstra

<http://www.azquotes.com/quote/561997>

An chuid is mó de na hearráidí i ríomhchláir, is féidir iad a rangú mar: earráidí comhréire, earráidí rite nó earráidí loighce.

Earráidí Comhréire

Tá earráidí comhréire giota beag cosúil leis na hearráidí litríthe, poncaíochta nó gramadúla a thiocfadh leat a dhéanamh agus tú ag scríobh teanga nádúrtha mar an Béarla.

Syntax Errors

“[Syntax errors] occur either when the program statements cannot be understood because they do not follow the rules laid down by the programming language, *statement syntax errors*, or when program structures are incorrectly formed, *program syntax errors* or *structure errors*.”

BCS Glossary of Computing, 13th edition, p 301.

Earráidí Rite

Cé gur féidir earráidí comhréire a aimsiú go minic roimh nó le linn tiomsúcháin, ní thagann roinnt earráidí chun solais go dtí go ndéantar an ríomhchlár a rith. Tugtar *execution errors* nó *run-time errors* orthu seo.

Execution Errors

“[Execution errors] or run-time errors are errors that are detected during program execution.

These errors, such as overflow and division by zero, can occur if a mistake is made in the processing algorithm or as a result of external effects not catered for by the program, such as lack of memory or unusual data.”

BCS Glossary of Computing, 13th edition, p 301.

Earráidí Loighce

Is é is earráid loighce ann, earráid i ndearadh ríomhchláir a iompraítear ar aghaidh ansin sa chód. Ríomhchlár a bhfuil earráid loighce ann, is gnách go dtiomsóidh sé agus go rithfidh sé gan aon leid go bhfuil an earráid ann. Agus sin ráite, ní tháirgfidh sin an toradh ceart.

Logic Errors

“[Logical errors] are mistakes in the design of a program, such as the use of an inappropriate mathematical formula or control structure, recognised by incorrect results or unexpected displays.”

BCS Glossary of Computing, 13th edition, p 301.

Craic le Dífhabhtú!

Is áireamhán simplí é an ríomhchlár gairid Python seo a leanas. Tá sé de rún aige go spreagfadh sé an t-úsáideoir le dhá uimhir agus oibríocht a thabhairt, an oibríocht a chur i bhfeidhm ar na huimhreacha agus an toradh a phriontáil. Roghnaítear na hoibríochtaí ó: suimiúchán (*add*), dealú (*sub*), iolrú (*mul*) agus roinnt (*div*).

```
n1 = input('Type first number: ')
n2 = input('Type second number: ')
op = input('Type an operation [add, sub, mul or div]: ')
if op == 'add':
    n = n1 + n2
elif op == 'sub':
    n = n1 - n2
elif op == 'mul':
    n = n1 * n2
else op == 'div':
    n = n1 / n2
print(N)
```

Má dhéanann tú iarracht an ríomhchlár seo a rith ina fhoirm reatha, tabharfaidh Python rabhadh duit faoi earráid chomhréire sa choinníoll ar an chéad chéim den ráiteas roghnaithe (*op='add'*). Is féidir le Python a léiriú go bhfuil earráid ann ach ní féidir leis do chód a cheartú – seo obair don ríomhchláraitheoir. Sa chás seo, tá an t-oibrítheoir sannacháin (=) úsáidte againn nuair ba chóir dúinn an t-oibrítheoir comparáide a úsáid (==). Tá sé seo ceartaithe thíos.

```
n1 = input('Type first number: ')
n2 = input('Type second number: ')
op = input('Type an operation [add, sub, mul or div]: ')
if op == 'add':
    n = n1 + n2
elif op == 'sub':
    n = n1 - n2
elif op == 'mul':
    n = n1 * n2
else op == 'div':
    n = n1 / n2
print(N)
```

Má dhéanann tú iarracht an leagan seo a rith, arís eile, tabharfaidh Python rabhadh duit go bhfuil earráid chomhréire ann – an iarraidh seo, sa chéim deiridh den ráiteas roghnaithe (*op == 'div'*). An fáth gur earráid é seo ná go dtosaíonn an coinníoll seo leis an eochairfhocal *else*, ach níor chóir coinníoll a bheith ann i ndiaidh *else*. Más mian leat coinníoll air seo, ní mór duit úsáid a bhaint as an eochairfhocal *elif*. Tá sé seo ceartaithe thíos.

```
n1 = input('Type first number: ')
n2 = input('Type second number: ')
op = input('Type an operation [add, sub, mul or div]: ')
if op == 'add':
    n = n1 + n2
elif op == 'sub':
    n = n1 - n2
elif op == 'mul':
    n = n1 * n2
elif op == 'div':
    n = n1 / n2
print(N)
```

Má dhéanann tú iarracht an leagan seo den ríomhchlár a rith, rachaidh tú giota beag níos faide an iarraidh seo. Seo idirghníomhú samplach.

```
Type first number: 2
Type second number: 3
Type an operation [add, sub, mul or div]: add
Traceback (most recent call last):
  File "d:\Users\Norman\Desktop\temp.py", line
  15, in <module>
    print(N)
NameError: name 'N' is not defined
>>>
```

Nochtar na leideanna agus is féidir leat roinnt sonraí a chlósriobh isteach ach ansin tarlaíonn earráid. Tá tuairisc ar earráidí Python taispeánta i gcló trom.

Tugtar *NameError* air seo: tá an ríomhchlár ag iarraidh athróg (N) a phriontáil nár sannadh luach di. Ag an pointe seo, ba chóir go dtuigfeadh an ríomhchláraitheoir/dífhabhtóir go bhfuil N (cás uachtair) úsáidte aici nuair ba chóir di n (cás íochtaire) a úsáid. Tá sé seo ceartaithe thíos.

```
n1 = input('Type first number: ')
n2 = input('Type second number: ')
op = input('Type an operation [add, sub, mul or
div]: ')
if op == 'add':
    n = n1 + n2
elif op == 'sub':
    n = n1 - n2
elif op == 'mul':
    n = n1 * n2
elif op == 'div':
    n = n1 / n2
print(n)
```

Rithfidh an leagan seo den ríomhchlár anois gan aon earráidí a thógáil. Mar shampla:

```
Type first number: 2
Type second number: 3
Type an operation [add, sub, mul or div]: add
23
>>>
```

Ar an drochuair, tá fadhb ann go fóill leis an chód – ba chóir gur 5 an toradh – ní 23! Ríomhchláraitheoir/dífhabhtóir a bhfuil taithí aige, tuigfidh sé go bhfuil an earráid ann mar gheall go dtugann an fheidhm, *input*, teaghrán ar ais agus ní cineál uimhriúil. Dá thoradh sin, déanann an t-oibrítheoir '+' comhchaitéiniú teaghráin in áit suimiúcháin uimhriúla. Déantar seo a cheartú thíos.

```
n1 = int(input('Type first number: '))
n2 = int(input('Type second number: '))
op = input('Type an operation [add, sub, mul or
div]: ')
if op == 'add':
    n = n1 + n2
elif op == 'sub':
    n = n1 - n2
elif op == 'mul':
    n = n1 * n2
elif op == 'div':
    n = n1 / n2
print(n)
```

Bunaithe ar an idirghníomhú seo a leanas, seans go mbeadh cathú ort smaoineamh go bhfuil an ríomhchlár saor ó earráid anois...

```
Type first number: 2
Type second number: 3
Type an operation [add, sub, mul or div]: add
5
>>>
```

... ach ní bheadh an ceart agat! Ní mór don tástáil éifeachtach úsáid a bhaint as raon de shonraí ionchuir, agus gach conair trí ríomhchlár a thástáil. Cad é a tharlóidh, mar shampla, má dhéanann an t-úsáideoir earráid chlósriobhneoireachta?

```
Type first number: 2
Type second number: 3
Type an operation [add, sub, mul or div]: dib
Traceback (most recent call last):
  File "d:\Users\Norman\Desktop\temp.py", line
  15, in <module>
    print(n)
NameError: name 'n' is not defined
>>>
```

San idirghníomhú seo, chlósriobh an t-úsáideoir 'dib' in áit 'div'. Ós rud é nach bhfuil sé ag teacht le ceann ar bith de chéimeanna an ráitis roghnaithe, téann an rialúchán ar aghaidh chuig an ráiteas prionta gan luach a bheith sannta do n – sin cúis na hearráide. Is féidir an cód a athstruchtúrú leis an earráid seo a cheartú ach dóigh níos fearr ná láimhseáil eisceachtaí Python a úsáid. Déantar cur síos air seo sa chéad rannán eile.

Láimhseáil Eisceachtaí

Is *execution error* é *NameError* in Python – ar a dtugtar *exception* in amanna. Nuair a tharlaíonn earráid mar seo, deir muid *that an exception has been raised*. Is píosa cóid é *exception handler* atá deartha leis na hearráidí seo a cheapadh.

An bealach is simplí le déileáil leis an eisceacht seo ná comhstruchtúr *try-except* Python a úsáid. An comhstruchtúr seo, insíonn sé do Python *try* a dhéanamh le tasc a chríochnú ach chomh maith leis sin sainíonn sé cód malartach atá gairthe má tharlaíonn earráid. An ríomhchlár thíos, cuimsíonn sé láimhseálaí eisceachtaí don *NameError*.

```
n1 = int(input('Type first number: '))
n2 = int(input('Type second number: '))
op = input('Type an operation [add, sub, mul or div]: ')
if op == 'add':
    n = n1 + n2
elif op == 'sub':
    n = n1 - n2
elif op == 'mul':
    n = n1 * n2
elif op == 'div':
    n = n1 / n2
try:
    print(n)
except NameError:
    print('Operation not recognised.')
```

Is é éifeacht an ráitis *try-except* ná n a phriontáil ar an dóigh normalach más amhlaidh nach bhfuil aon earráidí ann. Agus sin ráite, má bhíonn *NameError* ann, déantar an teaghrán *'Operation not recognised.'* a phriontáil ina áit sin.

Mar sin de – an bhfuil an ríomhchlár ceart anois? Bhuel, smaoinigh ar an idirghníomhú seo:

```
Type first number: 2
Type second number: 0
Type an operation [add, sub, mul or div]: div
Traceback (most recent call last):
  File "d:\Users\Norman\Desktop\temp.py", line
  13, in <module>
    n = n1 / n2
ZeroDivisionError: division by zero
```

An iarraidh seo tógar eisceacht cionn is go bhfuil an ríomhchlár ag iarraidh roinnt ar náid. Thig linn seo a réiteach trí láimhseálaí eisceachtaí eile a chruthú.

```
n1 = int(input('Type first number: '))
n2 = int(input('Type second number: '))
op = input('Type an operation [add, sub, mul or div]: ')
try:
    if op == 'add':
        n = n1 + n2
    elif op == 'sub':
        n = n1 - n2
    elif op == 'mul':
        n = n1 * n2
    elif op == 'div':
        n = n1 / n2
    print(n)
except NameError:
    print('Operation not recognised.')
except ZeroDivisionError:
    print('Cannot divide by zero.')
```

Rinneadh dhá athrú anseo. Is é an ceann is soiléire ná gur cruthaíodh láimhseálaí nua eisceachtaí le déileáil le hearráidí a bhaineann le roinnt ar náid. An ceann nach bhfuil chomh soiléir céanna ná gur bogadh an eochairfhocal, *try*, chuig pointe níos luaithe sa ríomhchlár. Seo cionn is nach n-oibríonn an comhstruchtúr *try-except* ach amháin d'eisceachtaí atá tógtha taobh istigh den rannán *try*.

Ceist: An bhfuil an ríomhchlár ceart anois?

Ceist: Cad é mar atá a fhios agat?

Áiseanna

- BBC Bitesize, Writing Error-Free Code: <http://www.bbc.co.uk/education/guides/zcxfyrd/revision>
- TutorialsPoint, Python 3 Tutorial: <https://www.tutorialspoint.com/python3/index.htm>
- TutorialsPoint, Python 3 – Exception Handling: https://www.tutorialspoint.com/python3/python_exceptions.htm

