Python
Switches / flags
indicators / boolean

Uses

Set indicators that can be tested later.

The indicators can be, but not limited to:

- Switches
- Flags
- Boolean
- Codes
- Counters
- message
- etc

You already using an indicators

Run the program again

Using either

continue (y/n)

Or

1 to continue or 0 to quit

Example program 1 of 3

```
## input routine
def input1(msg):
    indata = input(msg)
    return indata
## Processing
def process1():
    msg = "Enter a number: "
    num = input(msg)
    (sw1,b) = valid1(num)
    return (sw1,b)
```

Note: sw1

Example program
2 of 3

```
##  validation
def valid1(num):
    err = 0
    b = 0
    try:
        b = float(num)
        if (b >0):
            err = 0
        else:
            err = 1
    except:
        err = 1
    return (err,b)
```

NOTE

Example Program 3 of 3

```
## main line
def main():
    ans = 'y'
    total = 0
    while(ans.upper() == 'Y'):
        (sw1,num) = process1()
        if (sw1 == 0):
            print('Number: ',num)
            total = total + num
        else:
            print(' ***** Invalid INPUT')
        msg = 'Continue (y/n)? '
        ans = input1(msg)
    print('-------')
    print('total is: ', total)
    return
main()
```

NOTE

Using Boolean

Define a variable and set to either true or false

Example

ind1 = True

ind2 = False

Note: true with a capital T

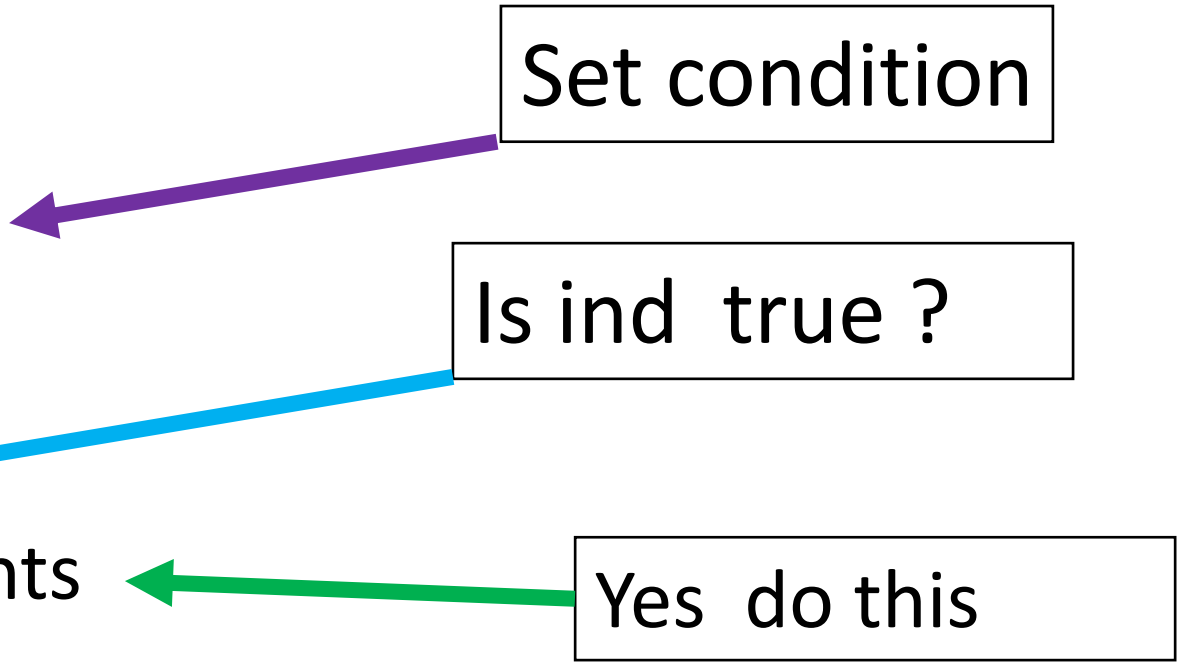
false with a capital F

Using Boolean Example

Example

```
ind = True

if (ind):
    statements
else:
    statements
```



Changes to previous program

```
##  validation

def valid1(num):
    err = True
    b = 0
    try:
        b = float(num)
        if (b >0):
            err = True
        else:
            err = False
    except:
        err = False
    return (err,b)
```

Change to main

```
if (err):
        print('Number: ',num)
        total = total + num
```

Using codes

Define an error code variable.

Put codes of either numeric or alpha to indicate error level.

Example

Code list

0 = all good

1 = input empty

2 = not a number

error = 0

Change above variable reflecting error detected

Changes to program

```
##  validation
def valid1(num):
    err = 0
    b = 0
    try:
        b = float(num)
        if (b >0):
            err = 0
        else:
            err = 1
    except:
        err = 2
    return (err,b)
```

Changes in main

```
if (err==0):
        print('Number: ',num)
        total = total + num
    else:
        if err ==1:
            print(' input cannot be negative')
        else:
            print(' ***** Invalid INPUT')
```

done