

C: If statements

Primary method for making a script's execution depend on data values

1. Statement itself provides a **logical test**
2. Followed by an **indented block** of code executed if the test is true

Many, many logical tests are available:

<code>a == b</code>	Is a the same as b ?
<code>a != b</code>	Are a and b different?
<code>a > b</code>	Is a greater than b ?
<code>"me" in "something"</code>	Is string "me" in "something" ?
<code>filename.endswith("csv")</code>	Does the string in filename end with "csv" ?
<code>len(a_list) == 4</code>	Does a_list have 4 elements?
...	...

Test result will be **True** or **False**

Can be stored in a variable:

<code>short_word = len(x) < 5</code>	Is the length of the string in X less than 5
<code>blank = x.isspace()</code>	Does X contain only blank spaces?
<code>missing = x is None</code>	Is X set to None?

Examples of IF statements:

```
something = 10
```

<pre> if something == 10: print("ten") something = 20 if something == 10: print("ten") other = "some long string" if other.endswith("ring"): count= len(other.split()) reply = ["ding" for n in range(count)] print(" ".join(reply)) </pre>	<p>prints "ten"</p> <p>nothing prints</p> <p>prints what?</p>
--	---

More complex **if** statements can have **elif** ("else if") and **else** blocks

- **elif** statements have an additional **logical test**

Their blocks executed only if:

- previous statements are False, **and**
- the current test is True

- **else** statements have no logical test

Their blocks are executed if all previous tests are False

Example:

```

list1 = [30,10,20]

for item in list1:

```

```
if item == 10:
    print( "ten" )
elif item < 30:
    print( "bigger" )
else:
    print( "other" )
```

print "ten" if item is 10

print "bigger" if item is NOT 10 and is less than 30

print "other" if item is not 10 and not less than 30

Output:

Item	Printed output	Trigger
30	"other"	else clause
10	"ten"	if clause
20	"bigger"	elif clause

Used in g06, along with a number of string functions