It's about time

- To design interesting robot control programs requires detailed thinking about sequence and time.
- Problem: Find out how much time it takes for a computer to execute (i.e., do) basic program operations.
It's about time

Problem: Find out how much time it takes for a computer to execute (i.e., do) basic program operations.

Operation and any return value printing seem instantaneous when you press Enter.

Write: Your name + your best choice of a plan to solve this problem. Take 3 minutes. (I will collect and read your answers aloud, w/o names.)
Collected Ideas....

- (from prof & an anonymous student) Computer operations/programs ARE instantaneous. Scientists must think of that! Maybe...

- instantaneous for all intents and purposes

- just run

- observe feelings of frustration when computer doesn't respond when I press enter

- time execution of a “rather long program” because it will be easier to time
- use a timer, if available, that can measure short intervals of time
- build program with multiple operations and see if this takes more time than one simple operation
- enter a command that will show think time, or use some kind of screen timer.
- ask computer to perform a difficult task so the time taken is visible
- .. long program consisting of the same basic operation, let's say 1,000 times. Time the interval between the start and end and divide by 1,000.
- use a stopwatch
- write several different operations together, as one function (for example, an addition problem followed by a subtraction problem followed by a multiplication problem..) Run the function & measure the time. Then remove the functions one by one, measuring time of execution each occasion.
- go to ”My Computer” screen and find the hertz speed of your processor. [it] generally indicates how fast your computer can perform basic calc's. Generally it can do 0.5 million in a second.
write a program that can start whenever a operation begins and gives the "operation time" after each result–time from when "enter" is pressed to when the result appears on the screen.

- Problem is the timing itself would take time..........

- look at the system log. time would be in milliseconds and the only accurate judge of this time would be the computer itself.

- The time is variable depending on how complex the code is and how fast the computer is able to interpret the code.
- run function for a math problem. Then you have to print computing time

- Use the following command in a bash shell script (in UNIX): `time python pythoncode.py`
  - the Python code is saved in `pythoncode.py`

- when you save a function you can choose when you want it executed

- algorithm (not python code):

- Use a language's built-in time counter or wait time function...

  - this task will SKEW the overall time...

  ```python
  x = current_time
  y = y + 1
  z = current_time
  time_taken = z - y
  print time_taken
  ```
Towards a Counting Loop

- $i = i + 1$
Towards a Counting Loop

- 
- 
- 
- 
- $i = i + 1$
  - NameError: name 'i' is not defined
- 
- 
- 
- 
-
Towards a Counting Loop

- 
  - $i = 0$
- 
  - $i = i + 1$
Towards a Counting Loop

- $i = 0$

- $i = i + 1$

- print $i$
  
  1
Towards a Counting Loop

- i = 0
- while i < 10:
  - i = i + 1
  - print i
```python
>>> i = 0

Indentation...

>>> while i < 10:
    i = i + 1  # Automatically indents!

If you override that by keying backspace,

>>> while i < 10
    i = i + 1

File "<pyshell#84>", line 2
    i = i + 1
    ^

IndentationError: expected an indented block

>>> DON'T Panic  read the message to tell what is the problem, and edit/redo the coding with the problem fixed.
```
Towards a Counting Loop

- i = 0
- while i < 10:
  - print i
Towards a Counting Loop

- $i = 0$
- while $i < 10$:
  - print $i$  

(press Enter twice...)
HELP.. ): .... !?!???? What do I do!!??

Press Control-c
to interrupt an infinite loop
Press Control-c
to interrupt an infinite loop

Traceback (most recent call last):
  File "<pyshell#17>", line 2, in <module>
    print i
  File "/usr/lib/python2.5/idlelib/PyShell.py", line 1248, in write
    self.shell.write(s, self.tags)
  File "/usr/lib/python2.5/idlelib/PyShell.py", line 1237, in write
    raise KeyboardInterrupt
KeyboardInterrupt

>>>
Towards a Counting Loop

- 
  - i = 0
  - while i < 10:
    - print i
  - i = i + 1

Press Enter twice...
>>> i = 0
>>> while i < 10:
    print i
    i = i + 1

Press Enter twice...

0
1
2
3
4
5
6
7
8
9

See the chevron..
>>> i = 0
>>> while i < 10:
    print i
    i = i + 1

Now put cursor after "i = 0", press enter twice

>>> i = 0
2nd, put cursor after "i = i + 1", press enter once
>>> while i < 10:
    print i
    i = i + 1

You can now EDIT what you typed before!

>>> while i < 100:
Put cursor after i=i+1 again, enter 2x, new code runs.
>>> i = 0
>>> while i < 10:
    i = i + 1

Press Enter twice...

>>> i
10
>>> i = 0
>>> while i < 1000
>>>    i = i + 1

>>> i = 0
>>> while i < 1000000:
    i = i + 1
>>> i = 0
>>> while i < 10000000: # 10 million
    i = i + 1
    if i % 1000000 == 0: # 1 million
        print i

10000000
20000000
30000000
40000000
50000000
60000000
70000000
80000000
90000000
100000000

There is a short pause between the printing of successive lines.

It's too short to measure accurately with a stopwatch.

What can you do?