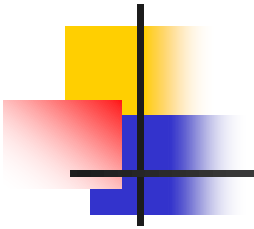


Introduction to Programming

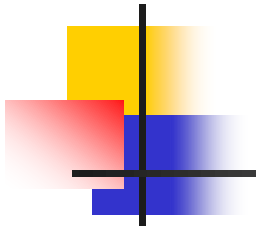
Department of Computer Science and Information Systems

Lecturer: Steve Maybank

sjmaybank@dcs.bbk.ac.uk

Autumn 2019 and Spring 2020

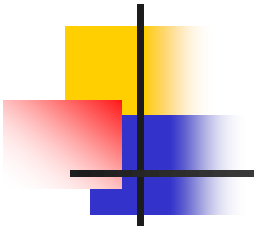
Week 2b: Review of Week 1, Variables



My First Program

```
# My first program
print("Hello World!")
```

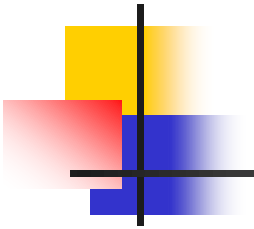
When the above program is run in IDLE the string "Hello World!" appears in the shell screen



Commentary

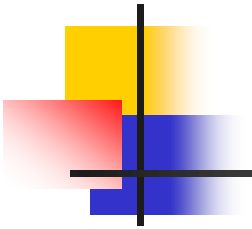
```
# My first program
print("Hello World!")
```

- The function print is called with the argument "Hello World!"
- The string "Hello World!" is written to the shell
- The statements within the function print are hidden
- The function print is in the Python Standard Library, PFE Appendix D



Strings

- A string is a sequence of characters, e.g. "Hello”.
- The quotes " " are a sign that a string is present. The quotes are not themselves part of the string.
  - What if we want to print " in a string? E.g., He said "yes".
  - print("He said \" yes\". ")
  - print('He said " yes". ')
- A string is not interpreted further, e.g. given "Hello" the compiler does not check to see if Hello is the name of a variable.



Errors

- Compile time errors: syntax errors found by the compiler, e.g.

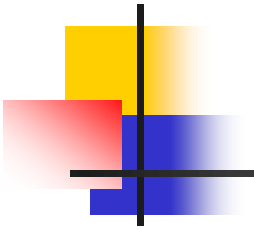
```
print)3)
```

- Run time errors (exceptions): errors which are not found by the compiler, but which prevent the program from running to completion, e.g.

```
print(1/(2-2))
```

- Run time errors (but not exceptions): the program compiles and runs but the output is not what is intended, e.g.

```
print("Hello Worrld!")
```



Investment Problem

- You put £10,000 into a bank account that earns 5% interest per year.
- How many years does it take for the account balance to be double the original?
- (PFE, Section 1.7)

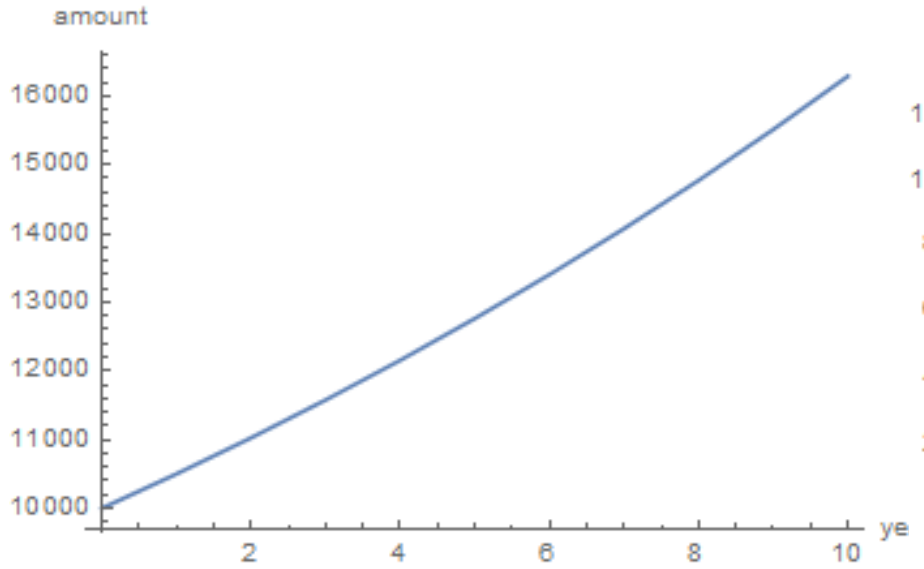
Solution to Investment Problem

- Initial balance:
  - £10000
- Interest rate:
  - 5% per year
- Interest earned after 1 year:
  - 10000*5/100 = 500
- Balance after 1 year:
  - initial balance + interest = 10000+500 = 10000*1.05
- Balance after two years:
  - 10000*1.05*1.05
- Balance after three years:
  - 10000*1.05*1.05*1.05
- Continue until the balance is
  - at least £20000

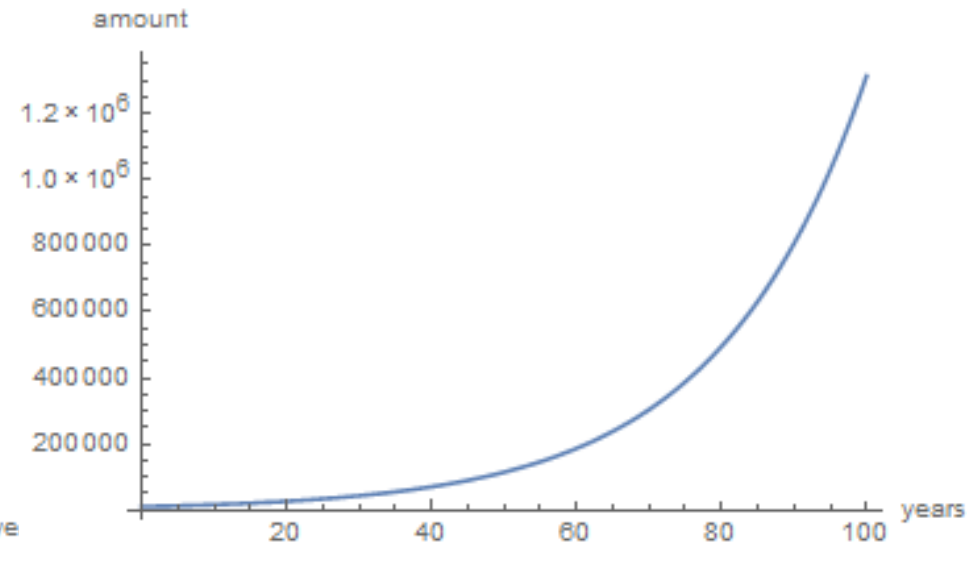
You put £10,000 into a bank account that earns 5% interest per year.

How many years does it take for the account balance to be double the original?

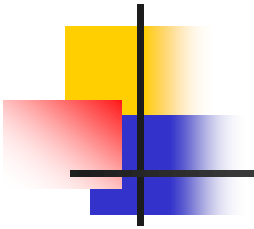
Graphs of the Balance



Graph for 10 years



Graph for 100 years



Algorithms

- An algorithm is a sequence of steps that is

  unambiguous

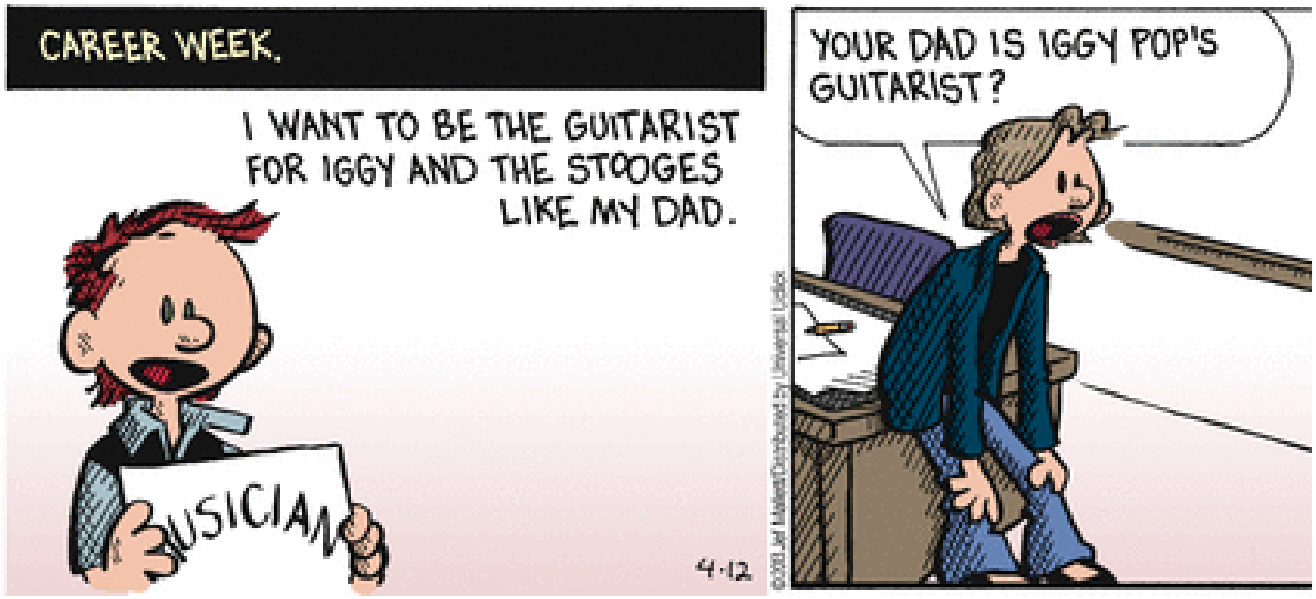
  executable

  terminating

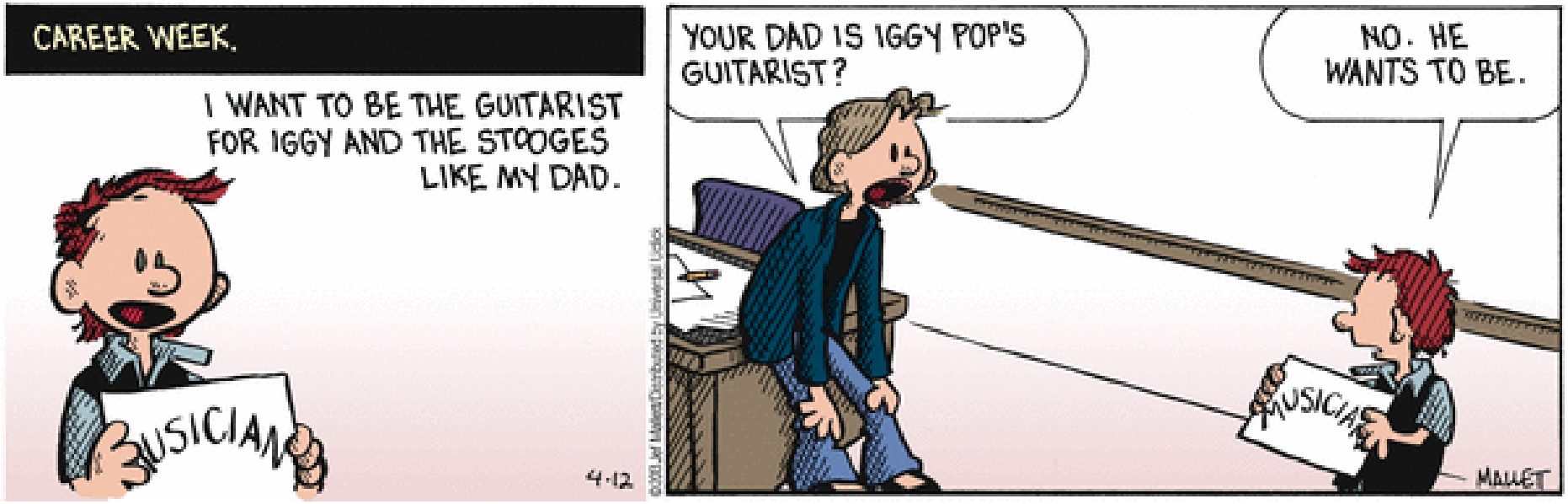
Ambiguity

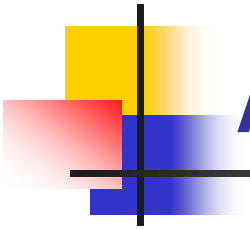

CAREER WEEK.
I WANT TO BE THE GUITARIST FOR IGGY AND THE STOOGES LIKE MY DAD.
MUSICIAN
4·12

Ambiguity


CAREER WEEK.
I WANT TO BE THE GUITARIST FOR IGGY AND THE STOOGES LIKE MY DAD.
MUSICIAN
4-12
YOUR DAD IS IGGY POP'S GUITARIST?

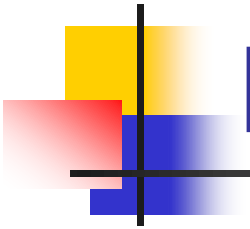
Ambiguity


CAREER WEEK.
I WANT TO BE THE GUITARIST FOR IGGY AND THE STOOGES LIKE MY DAD.
MUSICIAN
4-12
YOUR DAD IS IGGY POP'S GUITARIST?
NO. HE WANTS TO BE.
MUSICIAN
MALLET



Ambiguity

- Natural languages are not accurate
  - If it is cold, put on coat.
- Algorithms should be unambiguous
  - If it is less than 10 degrees, put on coat.



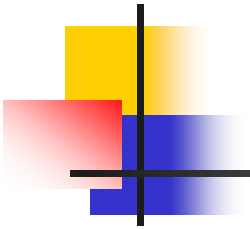
Executable

- A white flower
  - Nonexecutable!
- A statement has to do something
  - Pick a white flower

- Do the action for -2 times
  - Nonexecutable!
- Something that can be done by the program
  - Do the action for 2 times

Terminating

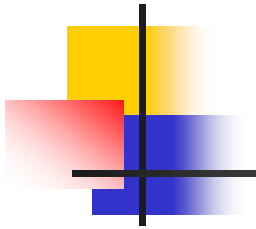
- The purpose of an algorithm is to deliver an answer to a problem.
- If you have to wait infinitely long to get the answer, it is less attractive.



Algorithms

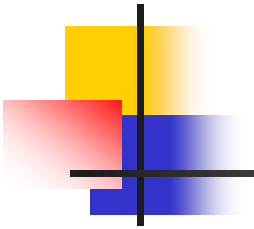
- An algorithm is a sequence of steps that is
  - unambiguous
  - executable
  - terminating
- The above pseudocode solution to the investment problem is an algorithm.
  - It terminates because the balance increases by at least £500 each year. Thus

    number of years <=(20000-10000)/500 = 20



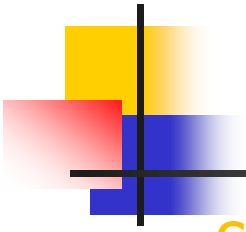
Program

```
print(10000*1.05*1.05*1.05)

# What does this line compute?

# Include additional factors 1.05 until a number greater
# than or equal to 20000 is printed.

# The strategy is crude but it works.
```



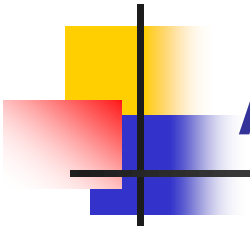
Variables

- A variable is a storage location in a computer program
- Each variable has a name and it holds a value
- Problem: does a six pack of 12 ounce drink cans contain more liquid than a two litre bottle?
- Appropriate names of variables:
  - cansPerPack
  - CAN_VOLUME
  - BOTTLE_VOLUME



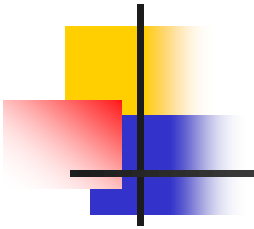
Assignment of a Value to a Variable

```
cansPerPack = 6  # assignment statement
# Left hand side: the name of a variable
# Right hand side: a value for the variable

print(cansPerPack)
# the value 6 of the variable cansPerPack will
# appear in the shell

cansPerPack = 8
# the previous value 6 is overwritten
```



Alternative Assignment Statement

```
cansPerPack = cansPerPack+2

# 1) Take the current value 8 of the variable cansPerPack
# 2) Evaluate the right hand side of the above statement:
#                 8+2 = 10
# 3) Assign the value 10 to the variable cansPerPack
```

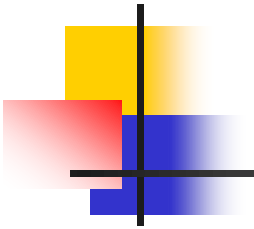


Creation of a Variable

If cansPerPack is used for the first time in a statement such as

```
cansPerPack = 6
```

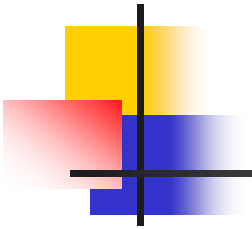
then the variable cansPerPack is created and initialised with the integer value 6.



Undefined Variables

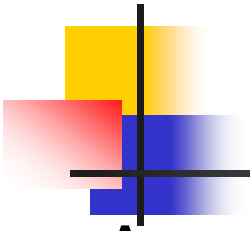
- A variable must be created and initialised before use.

```
print(cansPerPack)
# error if a value has not been assigned to cansPerPack

cansPerPack = 6
# cansPerPack is assigned a value but it is too late.
# The compiler does not look ahead
```



Number Types

- Number type: determines how a number is represented and the operations that can be carried out with that number.
- E.g. the int number type and the float number type.
- int: any whole number with no fractional part
  - e.g. -1, 0, 1
- float: any decimal fraction
  - e.g. -1.52, 3.4, - 9.400
  - e.g. 0.0, 2.0, -3.0
- Operations: addition, multiplication, division, etc.



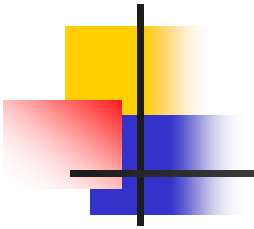
Number Literals

- A number literal is a number that appears explicitly in a program, e.g.

```
        q = 5    # What type is the value of q?
        # 5 is a number literal of type int
        q = 3.5 # What type is the value of q now?
        # 3.5 is a number literal of type float
# the value 5 is overwritten with the value 3.5 without error

        q = "test" # What type is the value of q now?
        # "test" is a string, not a number
# the value 3.5 is overwritten without error (not recommended)
```

Examples of Number Literals

Number	Type	Comment
6	int	An integer has no fractional part
-6	int	Integers can be negative
0	int	Zero is an integer
0.5	float	A number with a fractional part has type float
1.0	float	An integer with a fractional part .0 has type float
1E6	float	A number in exponential notation: $1*10^6$ or 1000000. Numbers in exponential notation always have type float.
2.96E-2	float	Negative exponent: $2.96 \times 10^{-2} = 2.96/100 = 0.0296$
100,000		**Error:** do not use a comma as a decimal separator
3 1/2		**Error:** Do not use fractions; use decimal notation: 3.5

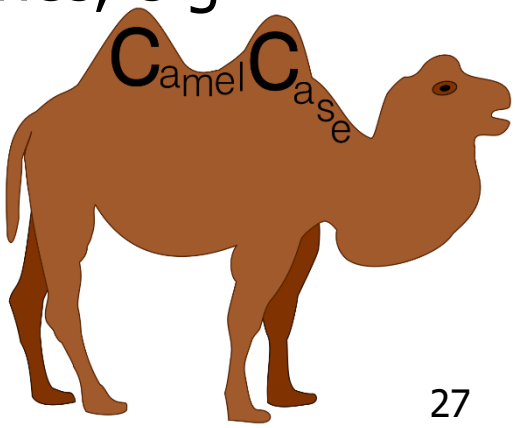


Names of Variables

- Names must start with a letter or underscore (_).
- The remaining characters must be letters, numbers or underscores
  - _____, 3letters, _3_3_3, rat^2, tot40_3, can volume
- Names are case sensitive
  - canVolume and canvolume
- Reserved words cannot be used, see PFE Appendix C
  - class, from, import, in, lambda, pass, return, with, yield, …

Recommended but not Obligatory

- If the value of the variable is significant and does not change, then use only capital letters and underscores in the name, e.g. BAKERS_DOZEN
- Otherwise, begin names of variables with a lower case letter, e.g. cansPerPack
- Use descriptive names, e.g. cansPerPack rather than cpp
- Use capital letters to mark word boundaries, e.g. cansPerPack – Camel naming



Names of Variables

Name of Variable	Comment
canVolume1	Names of variables consist of letters, numbers and underscores
x	Legal, but a more descriptive name is often better
CanVolume	Legal, but violates the convention that names of variables should begin with a lower case letter
6pack	**Error:** names of variables cannot start with a number
can volume	**Error:** names of variables cannot contain spaces
class	**Error:** names of variables cannot be reserved words
ltr/fl.oz	**Error:** symbols such as / or . cannot be used

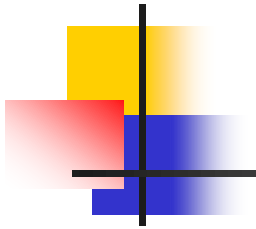
Review Questions

- R2.1. What is the value of mystery after this sequence of statements?

```
mystery = 1
mystery = 1-2*mystery
mystery = mystery+1
```

- R2.2. What is the value of mystery after this sequence of statements?

```
mystery = 1
mystery = mystery+1
mystery = 1-2*mystery
```



Compile Time Errors

- Cf. R2.8. Find at least three compile time errors in the following program

```
int x = 2
print(x, squared is, x*x)
xTripled = xDoubled + x
```