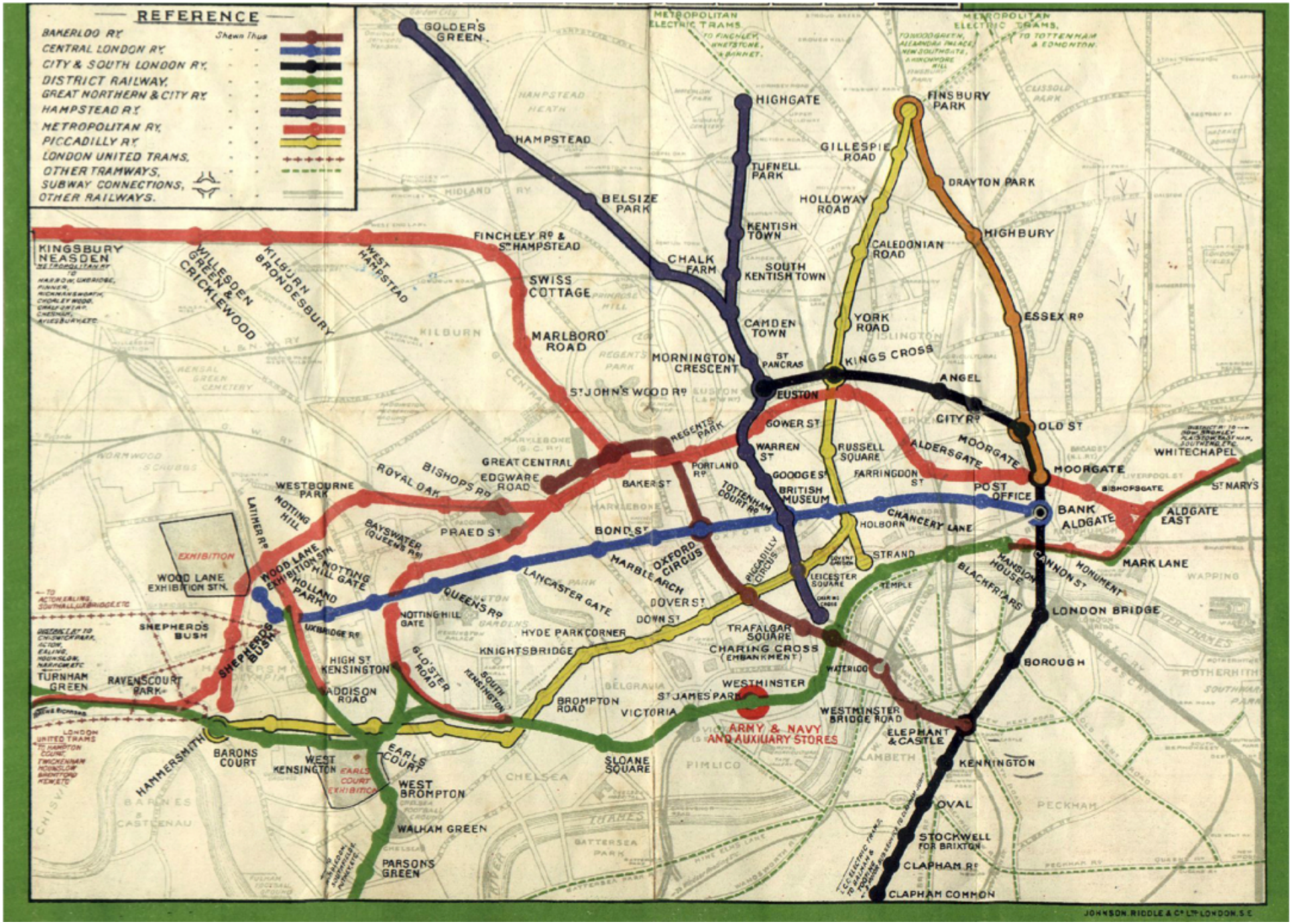
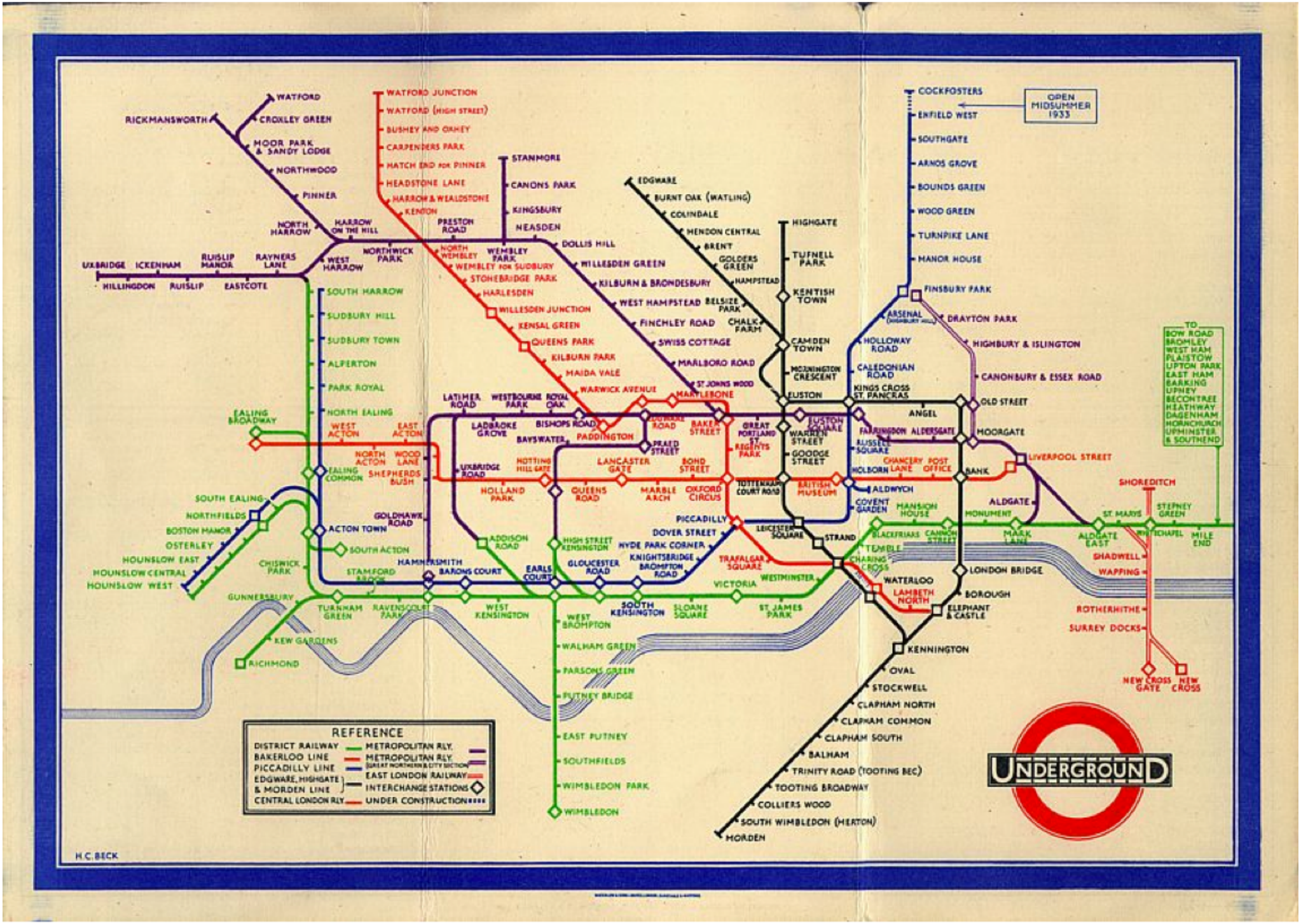
Algorithms, Abstraction, and Functions


REFERENCE
Shewn Thus
BAKERLOO RY.
CENTRAL LONDON RY.
CITY & SOUTH LONDON RY.
DISTRICT RAILWAY.
GREAT NORTHERN & CITY RY.
HAMPSTEAD RY.
METROPOLITAN RY.
PICCADILLY RY.
LONDON UNITED TRAMS.
OTHER TRAMWAYS.
SUBWAY CONNECTIONS.
OTHER RAILWAYS.
GOLDER'S GREEN
HAMPSTEAD
BELSIZE PARK
HIGHGATE
TUFNELL PARK
KENTISH TOWN
CHALK FARM
SOUTH KENTISH TOWN
CAMDEN TOWN
MORNINGTON CRESCENT
ST PANCRAS
KINGS CROSS
EUSTON
ANGEL
FINSBURY PARK
GILLESPIE ROAD
DRAYTON PARK
HOLLOWAY ROAD
CALEDONIAN ROAD
HIGHBURY
YORK ROAD
ESSEX Rd
CITY Rd
OLD ST
MOORGATE
WHITECHAPEL
ALDERSGATE
FINCHLEY Rd & St HAMPSTEAD
SWISS COTTAGE
MARLBORO' ROAD
ST JOHN'S WOOD Rd
KINGSBURY NEASDEN
WILLESDEN GREEN & CRICKLEWOOD
KILBURN BRONDESBURY
WEST HAMPSTEAD
REGENTS PARK
GOWER ST
WARREN ST
RUSSELL SQUARE
GREAT CENTRAL
EDGWARE ROAD
BISHOPS Rd
ROYAL OAK
PORTLAND Rd
GOODGES ST
FARRINGDON ST
BAKER ST
TOTTENHAM COURT Rd
BRITISH MUSEUM
POST OFFICE
BISHOPSGATE
ST MARY'S
ALDGATE EAST
BANK
ALDGATE
CHANCERY LANE
HOLBORN
WESTBOURNE PARK
NOTTING HILL
LATIMER Rd
BAYSWATER (QUEENS Rd)
PRAED ST
BOND ST
OXFORD CIRCUS
MARBLE ARCH
PICCADILLY CIRCUS
LEICESTER SQUARE
STRAND
MARK LANE
MONUMENT
CANNON ST
MANSION HOUSE
BLACKFRIARS
TEMPLE
LONDON BRIDGE
BOROUGH
EXHIBITION
WOOD LANE EXHIBITION STN.
WOOD LANE EXHIBITION STN.
HOLLAND PARK
NOTTING HILL GATE
LANCASTER GATE
QUEENS Rd
DOVER ST
DOWN ST
HYDE PARK CORNER
KNIGHTSBRIDGE
TRAFALGAR SQUARE
CHARING CROSS (EMBANKMENT)
WATERLOO
WESTMINSTER
ST JAMES' PARK
VICTORIA
BROMPTON ROAD
WESTMINSTER BRIDGE ROAD
ELEPHANT & CASTLE
KENNINGTON
OVAL
STOCKWELL FOR BRIXTON
CLAPHAM Rd
CLAPHAM COMMON
ARMY & NAVY AND AUXILIARY STORES
SHEPHERD'S BUSH
UXBRIDGE Rd
HIGH ST KENSINGTON
ADDISON ROAD
GLOUCESTER ROAD
SOUTH KENSINGTON
NOTTING HILL GATE
TURNHAM GREEN
RAVENSCOURT PARK
HAMMERSMITH
BARONS COURT
WEST KENSINGTON
EARLS COURT
EARLS COURT EXHIBITION
WEST BROMPTON
WALHAM GREEN
PARSONS GREEN
SLOANE SQUARE
CHELSEA
PIMLICO
LAMBETH
PECKHAM
BATTERSEA PARK
THAMES
HYDE PARK
KENSINGTON GARDENS
ROTHERHITH
WAPPING
SOUTHWARK PARK
JOHNSON, RIDDLE & Co. LTD. LONDON, S.E.


OPEN MIDSUMMER 1933
TO BOW ROAD BROMLEY WEST HAM PLAISTOW UPTON PARK EAST HAM BARKING UPNEY BECONTREE HEATHWAY DAGENHAM HORNCHURCH UPMINSTER & SOUTHEND
REFERENCE
DISTRICT RAILWAY
BAKERLOO LINE
PICCADILLY LINE
EDGWARE, HIGHGATE & MORDEN LINE
CENTRAL LONDON RLY
METROPOLITAN RLY.
METROPOLITAN RLY. (GREAT NORTHERN & CITY SECTION)
EAST LONDON RAILWAY
INTERCHANGE STATIONS
UNDER CONSTRUCTION
UNDERGROUND
H.C.BECK

Abstraction

"The essence of abstractions is preserving information that is relevant in a given context, and forgetting information that is irrelevant in that context."

John V. Guttag,
Introduction to Computation and Programming Using Python

Abstraction

*"The essence of abstractions is **preserving** information that is relevant in a given context, and **forgetting** information that is irrelevant in that context."*

John V. Guttag,
Introduction to Computation and Programming Using Python

- Think of a task or problem where:
  - you used to have to think about each step of the problem
  - but now it's automatic and you don't need to think about each step anymore.

Supercalifragilisticexpialidocious!
Even though the sound of it
Is something quite atrocious
If you say it loud enough
You'll always sound precocious
Supercalifragilisticexpialidocious!

Um diddle diddle diddle um diddle ay
Um diddle diddle diddle um diddle ay!

Because I was afraid to speak
When I was just a lad
My father gave me nose a tweak
And told me I was bad
But then one day I learned a word
That saved me achin' nose
The biggest word I ever heard
And this is how it goes: Oh!

Supercalifragilisticexpialidocious!
Even though the sound of it
Is something quite atrocious
If you say it loud enough
You'll always sound precocious
Supercalifragilisticexpialidocious!

Um diddle diddle diddle um diddle ay
Um diddle diddle diddle um diddle ay!

He traveled all around the world
And everywhere he went
He'd use his word and all would say
"There goes a clever gent"
When Dukes and maharajas
Pass the time of day with me
I say me special word and then
They ask me out for tea

Supercalifragilisticexpialidocious!
Even though the sound of it
Is something quite atrocious
If you say it loud enough
You'll always sound precocious
Supercalifragilisticexpialidocious!

Supercalifragilisticexpialidocious!
Even though the sound of it
Is something quite atrocious
If you say it loud enough
You'll always sound precocious
Supercalifragilisticexpialidocious!

Um diddle diddle diddle um diddle ay
Um diddle diddle diddle um diddle ay!

Because I was afraid to speak
When I was just a lad
My father gave me nose a tweak
And told me I was bad
But then one day I learned a word
That saved me achin' nose
The biggest word I ever heard
And this is how it goes: Oh!

Supercalifragilisticexpialidocious!
Even though the sound of it
Is something quite atrocious
If you say it loud enough
You'll always sound precocious
Supercalifragilisticexpialidocious!

Um diddle diddle diddle um diddle ay
Um diddle diddle diddle um diddle ay!

He traveled all around the world
And everywhere he went
He'd use his word and all would say
"There goes a clever gent"
When Dukes and maharajas
Pass the time of day with me
I say me special word and then
They ask me out for tea

Supercalifragilisticexpialidocious!
Even though the sound of it
Is something quite atrocious
If you say it loud enough
You'll always sound precocious
Supercalifragilisticexpialidocious!

Functions

- Programmers will use functions to give a ***name*** to a **section (block) of code**.
- Any time you want to run that block, you can use the name instead of retyping or copy-and-pasting.

Functions

- To use a function, we must ***define*** it first.

Defining a function

Gives your function a name so it can be run later

- Syntax:

```
def name():
    statement      # Notice how these
    statement      # lines are indented.
    statement      # This is how Python knows
    …              # where a function definition
                   # begins and ends.
```

Pick a name for your function that describes what it does! (Just like you pick variable names that describe what the variable holds.)

Defining a function

Gives your function a name so it can be run later

- Syntax:

```
def print_chorus():
    print("Supercalifragilisticexpialidocious!")
    print("Even though the sound of it")
    print("Is something quite atrocious")
    print("If you say it loud enough")
    print("You'll always sound precocious")
    print("Supercalifragilisticexpialidocious!")
```

Functions

- To use a function, we must ***define*** it first.
- After defining a function, to run the code inside, you ***call*** the function.

Calling a function

Runs the code inside the function definition

- Syntax:

```
name()
```

After defining a function, you can call it any number of times you want.

Each time it is called Python acts as if you had typed in all of the lines of the function definition.

- You are in charge of dessert for Thanksgiving dinner. You decide to make two pumpkin pies and an apple pie.
- Write a program that ***defines*** three functions:
  - `make_apple()` should print a description of how to make an apple pie.
  - `make_pumpkin()` should print a description of how to make a pumpkin pie.
  - `cook_dinner()` should ***call*** `make_apple()` and `make_pumpkin()` appropriately to make the pies.

The `main()` function

- Python programs usually include a `main()` function that is the first function that runs when the program begins.
  - This function is in charge of calling any other functions.
- This is not (technically) required in Python, but is a good habit.
  - Required in other languages like C++ and Java.
  - Required for CS 141! ☺

The `main()` function

- From this point on, always ***define*** a `main()` function in your programs.
- Always ***call*** the `main()` function as the last line of your program (your `.py` file).

```
def print_chorus():
    print("Supercali…")
    (etc)

def print_um_diddle():
    print("Um diddle diddle…")
    (etc)

def print_verse1():
    print("Because I was afraid to speak…")
    (etc)

# A function for the "main" program.
def main():
     print_chorus()         # Print the chorus
     print_um_diddle()      # Print the um diddles
     print_verse1()         # Print the 1st verse
     print_chorus()         # Print the chorus again
     print_um_diddle()      # Print the um diddles again
     print_verse2()         # Print the 2nd verse
     print_chorus()         # Print the chorus the last time

main()                      # Start the program
```

- When a function is called, Python will
  - "jump" to the first line of the function's definition,
  - run all the lines of code inside the definition, then
  - "jump" back to the point where the function was called.

```
1  def twinkle():
2    print("Twinkle twinkle little star")
3    print("How I wonder what you are")

4  def main():
5     twinkle()        # Call (run) the twinkle function.
6     print("Up above the world so high")
7     print("Like a diamond in the sky")
8     twinkle()        # Call the twinkle function again.

9  main()              # Call main() to start the program.
```