SO YOU WANT TO PYTHON

WHAT THIS TALK WILL NOT BE

» this talk will not be an exhaustive feature tour

» I will not teach you syntax

» you won't even be able to write python by the end of the talk

WHAT I AM LOOKING TO DO

» I want to show you a few problems and how we solve them in python.

» I want to give you a few hooks so that you can get started with learning it , if you are interested.

» I want to expose you to the core principles and ideologies of python…

» and the general culture of python

IMPORT THIS?

THE ZEN OF PYTHON

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.

In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than right now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.

QUICK SURVEY

META

» There will be a gradual ramp in difficulty

» I will get geeky a few times

» Each problem will expose something new

» The idea is not to learn how to write python but understand what the point of this language is.

» there will be code but you don’t have to understand it , just take away the ideology.

EXCHANGING 2 VALUES

THE C WAY

```c
main() {
int a = 10,b = 20,temp;

temp = a;
a = b;
b = temp;

}
```

ISN'T THIS HIDEOUS?

» An extra variable.

» Type declarations.

» Syntactic noise.

THERE MUST BE A BETTER WAY!

“A HIGH LEVEL LANGUAGE IS ONE WHICH DOESN’T REQUIRE ATTENTION TO THE IRRELEVANT.”

LET'S LOOK AT THE PYTHONIC WAY TO DO THINGS

```
x = 20 , y = 10
x,y = y,x
```

READABILITY COUNTS , BEAUTIFUL IS BETTER THAN UGLY,
SIMPLE IS BETTER THAN COMPLEX

CONCEPT

» something that corresponds to a single unit of thought should be expressed in a single line , one logical step should be one LOC

» Readability counts a lot . Code is READ a lot more than it is WRITTEN

LETS TAKE THIS IDEA FOR A SPIN

```
def fib(n):
    x,y = 1,1
    for i in range(n-2):
        x,y = y,x+y
    return y
```

» Good code should be easy to read and in a way self documenting.

» Good code makes its meaning clear.

REVERSE A STRING

THE C WAY

THERE MUST BE A BETTER WAY!

THE PYTHONIC WAY

```
s = “this”

print(s[::-1])
```

CONCEPT

» Integrate powerful tools directly into the language

» Actively enforce good programming practices

» Higher level tools allow you to think on a more abstract level

» Learning to use the tools will be harder but you can be sure that they are the state of the art

LETS TAKE THIS IDEA OUT FOR A SPIN

```
def pali(s):
    if s == s[::-1] :
        return True
    else: return False
```

WHY DOES THIS MATTER ?

» programming isn't about solving small problems like this , instead it is about building systems and systematic solutions to problems…

» Problems much more complex than reversing a string…

» But all problems can be broken down into really small sub problems that rely on really simple pieces of code just like the one we saw.

» Not having to wrestle with the language allows us to think more clearly about the bigger picture.

EASY THINGS SHOULD BE EASY

LET ME SHOW YOU WHAT I MEAN.

```
def pali(s):
    if s == s[::-1] :
        return True
    else: return False


with open('foo.txt') as f:
    rv = {i.strip() for i in f if pali(i.strip())}
    print(rv)
```

-I just fed this code a list of ~497000 most common words on Reddit.
-It took me about 2 mins to write this code and about 10 mins to grab the words using PRAW.
-It ran in 0.2 seconds.

In just a fifth of a second i know the most common palindromes on Reddit.

```
= RESTART: /Users/govind/Documents/projects/python projects /palicounter.py =
{'CIC', 'sememes', 'tyt', 'kinnikinnik', 'AA', 'AAAA', 'AJA', 'AAAAAA', 'ecce', 'dewed', '
MFM', 'peeweep', 'RAR', 'FSF', 'i', 'ihi', 'repaper', 'AMA', 'JJ', 'SVS', 'APA', 'STS', 'm
inim', 'tenet', 'ottetto', 'AOA', 'oho', 'cyc', 'mesem', 'mmmm', 'ADA', 'denned', 'RSR', '
F', 'stats', 'TPT', 'hagigah', 'heh', 'CCC', 'C', 'y', 'OEO', 'PSP', 'ECE', 'CNC', 'non',
'EOE', 'E', 'ii', 'SSTTSS', 'eme', 'CFC', 'ABA', 'degged', 'kakkak', 'ette', 'G', 'n', 'dt
d', 'MRSRM', 'deled', 'sees', 'LCCL', 'tibbit', 'xxx', 'TST', 'K', 'LWL', 'MWM', 'kelek',
'ERE', 'pullup', 'CPC', 'shahs', 'SS', 'OSO', 'LAL', 'o', 'BB', 'tat-tat', 'ARA', 'CMC', '
iii', 'a', 'retter', 'U', 'hah', 'CC', 'halalah', 'maam', 'PIP', 'ese', 'CBC', 'esse', 'SV
VS', 'mem', 'HSH', 'oooo', 'waw-waw', 'SAS', 'hallah', 'marram', 'SOS', 'o-o', 'T', 'ICI',
'LBL', 'semes', 'X', 'RNR', 'ASA', 'malam', 'TCT', 'UU', 'txt', 'VAV', 'DFD', 'IDI', 'OBO'
, 'TFT', '-i-', 'AEA', 'eke', 'B', 'LPL', 'yoy', 'DID', 'lemel', 'prp', 'PUP', 'mom', 'rot
or', 'CDC', 'CEC', 'pip-pip', 'DCD', 'NPN', 'tipit', 'MIM', 'BBB', 'deked', 'ISI', 'redder
', 'reviver', 'SBS', 'tet', 'oxo', 'ana', 'civic', 'kook', 'MTM', 'aga', 'CTC', 'PNP', 'su
ccus', 'COC', 'CWC', 'waw', 'ALA', 'WW', "ma'am", 'tot', 'HRH', 'addda', 'OCO', 'AFA', 'go
og', 'NTN', 'OO', 'PDP', 'GBG', 'MMM', 'neven', 'LDL', 'D', 'CRC', 'TAT', 'mallam', 'TT',
'WSW', 'AAA', 'DSD', 'MGM', '-o-', 'rever', 'B/B', 'LL', 'SSS', 'refer', 'SWS', 'boob', 'p
eep', 'PVP', 'RMR', 's', 'V', 'toot', 'xix', 'tut-tut', 'RR', 'OAO', 'selles', 'kaiak', 'O
GO', 'noon', 'SLS', 'sooloos', "s's", 'eye', 'SRS', 'tebbet', 'W', 'RTR', 'J', 'ATA', 'tgt
', 'SCCS', 'SPS', 'WNW', 'trt', 'deedeed', 'KKK', 'DDD', 'MM', 'keek', 'ene', 'm', 'terret
', 'xx', 'Z', 'ELLE', 'imi', 'IPI', 'SES', 'SXS', 'PLP', 'PEP', 'PP', 'alula', 'affa', 'ya
y', 'DRD', 'acca', 'awa', 'sagas', 'VV', 'DAD', 'AKA', 'CAC', 'PAP', 'TWT', 'DMD', 'R', 'W
OW', 'TNT', 'level', 'PPP', 'gig', 'umu', 'murdrum', 'yaray', 'TSST', 'ululu', 'wow-wow',
'SDS', 'solos', 'tat-tat-tat', '2', 'FF', 'YY', 'huh', 'madam', 'teet', 'RADAR', 'ZZ', 'DO
D', 'AHA', 'kayak', 'tirrit', 'torot', 'CSC', 'L', 'HIH', 'deed', 'divid', 'PTP', 'Q', 'MS
M', 'usu', 'EEE', 'POP', 'HH', 'SCS', 'LCL', 'SMS', 'ISSI', 'AIA', 'GG', 'AUA', 'DD', 'ZZZ
', 'P', 'rotator', 'immi', 'sis', 'UPU', 'tkt', 'CAMAC', 'LTL', 'mum', 'sexes', 'siris', '
RSFSR', 'deified', 'mym', 'reifier', 'CTTC', 'H', 'ee'}
0.24309396743774414
```

PYTHON IS A TRUE HIGH LEVEL LANGUAGE

- Build small things first and then combine them
- Python helps you out at every stage by giving you amazingly well designed tools.
- You shouldn’t have to worry about the nitty-gritty.
- Works with you , it is amenable to various styles
  - functional
  - imperative

HAVE NO ILLUSIONS

» Good tools does not equal good artist.

» It is arguable that higher level tools are really hard to understand and to gain mastery with.

» Python cannot make you a good programmer. A good Java programmer will still write superior code

» Truly beautiful python is more of an art than a science.

WITH GREAT POWER COMES GREAT RESPONSIBILITY.

NOW FOR THE FUN STUFF

THE SECRET SAUCE OF PYTHON

» The amazing standard library

» The amazing third party libraries

» And above all the amazing community

LET'S SEND AN EMAIL

```
import smtplib

server = smtplib.SMTP('smtp.gmail.com', 587)
server.starttls()
server.login("fakeemail@fakecdf", "fakepass")

msg = " you know nothing jon snow "
server.sendmail("ygritte@wildling.com ", " jonsnow@kinginthenorth.com", msg)
server.quit()
```

That is all it takes to send an email with python (not really a very recent gmail change now requires you to get an OAuth token. That is a few more lines)

LET'S SUBSCRIBE TO SUBREDDITS

LET'S WRITE AN OBJECT DETECTION APP