Amuse Conf 2018

DASHBOARD
VISION

Main topics

Animation

Voice interfaces

Bots

Research

The design process

Animation

Show what happened and what can happen next

Focus the user's attention

Support navigation

Let the user opt out

Be natural (ease in & out)

Communicate ideas and be creative

Voice interfaces

It always triggers an emotional response

…even when people know that it's a bot

Different voices can be evaluated differently

…even when they give the exact same answer

DASHBOARD VISION

Bots

Make it clear, that it’s not a human

…but people could still react to it as it were

It can answer simple questions

…and relay more complex ones to a human operator

Offer choices

Be very careful with canned responses

How would you describe that to your grandmother?

My grandmother is dead 😞

Great, thank you for your answer! 😊👍

Research

Data vs insight

The design process

Focus on the process, not on the outcome

(Over)communication

Artefacts (you won’t be there all the time)

Form & communicate opinions

Broaden you scope – zoom in, out & upwards

Learn all the skills you need & some more

Emotions & humanity

Fun vs utilitarian interfaces

UX writing & microcopy

Design for attention

Python intro

DASHBOARD
VISION

Basics

Significant white space

```
def holy_hand_grenade():
    return [1, 2, 5]
```

Formatting can still be custom if it’s consistent

Basics

Object-oriented (but not strictly)

Extensive standard library

Extensions can be written in C

Flow control

```
if parrot ≠ 'alive':
    return 'it’s a stiff!'

for s in ['spam', 'egg', 'sausage', 'spam']:
    print(s)

for i in range(42):
    print(i)
```

Flow control

Break, for-else

```
for s in ['spam', 'spam', 'spam']:
    if s ≠ 'spam':
        print('anything without spam')
        break
else:
    print('wonderful spam!')
```

Data structures

Lists

```
knights = ['Sir Lancelot', 'Sir Galahad', 'Sir Robin']
print(knights[-2:])

['Sir Galahad', 'Sir Robin']
```

Data structures

Tuples

```
our_weapons = 'surprise', 'fear', 'ruthless
efficiency', ('spam', 'spam'), 42
```

Tuples are immutable

Data structures

Sets

```
menu = {'spam', 'egg', 'sausage', 'spam'}
print(menu)

{'egg', 'spam', 'sausage'}
```

Unordered collection with no duplicate elements

Data structures

Dictionaries

```
knight = {
    'name': 'Sir Lancelot',
    'quest': 'to seek the Holy Grail',
    'favorite_color': 'blue'
}
print(knight['name'])

'Sir Lancelot'
```

Functions

Defining

```
def parrot(expired, type='late'):
    if expired:
        print(f'This is a {type} parrot!')
    else:
        pass
```

Keyword arguments must follow positional arguments

Functions

Variadic functions

```
def func(x, *args, **kwargs):
    pass

def concat(*args, sep='/'):
    return sep.join(args)
```

'sep' is mandatory and keyword-only

Functions

Lambda functions

```
def make_incrementor(n):
    return lambda x: x + n

f = make_incrementor(42)
print(f(0))
42
print(f(1))
43
```

Functions

Decorators

```
def double(func):
    def wrapper():
        func()
        func()
    return wrapper

@double
def run():
    print('Run away!')
```

Looping

List comprehensions

```
print([s.upper() for s in ['spam', 'spam', 'wonderful spam']])

['SPAM', 'SPAM', 'WONDERFUL SPAM']
```

Looping

Nested list comprehensions

```
matrix = [
    [1, 2, 3, 4],
    [5, 6, 7, 8],
    [9, 10, 11, 12],
]
print([[row[i] for row in matrix] for i in range(4)])

[[1, 5, 9], [2, 6, 10], [3, 7, 11], [4, 8, 12]]
```

Looping

Dictionary comprehensions

```
print({n: n**2 for n in range(5)})

{0: 0, 1: 1, 2: 4, 3: 9, 4: 16}
```

Modules

Import

```
import random

from random import choice, shuffle
```

Don’t do this

```
from random import *
```

Classes

Defining

```
class DerivedClass(Base1, Base2):

    def __init__(self, x):
        self.x = x
```

No private variables, but the ones starting with an underscore are not supposed to be referenced from outside

A class can also be empty

Environment

Virtual environment

```
python3 -m venv .ve
source .ve/bin/activate
```

Environment

PIP

Package management system

```
pip install 'SomePackage≥1.0.4'

pip freeze > requirements.txt

pip install -r requirements.txt
```

Resources

http://docs.python.org

http://www.learnpython.org

https://realpython.com

https://docs.python-guide.org