

# Introduction à l'analyse d'images 2D

*Automne 2022 - cours 2*



UNIVERSITÉ  
DE GENÈVE

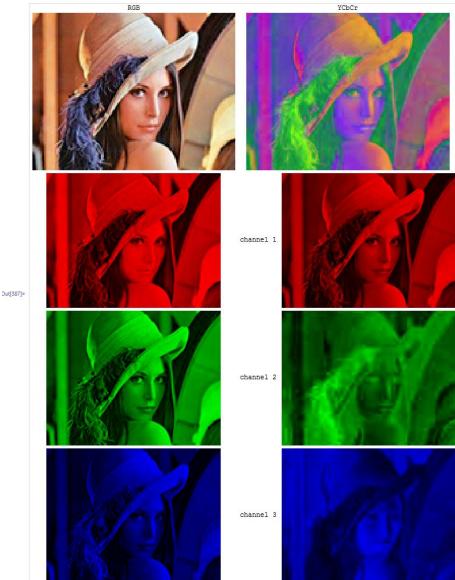
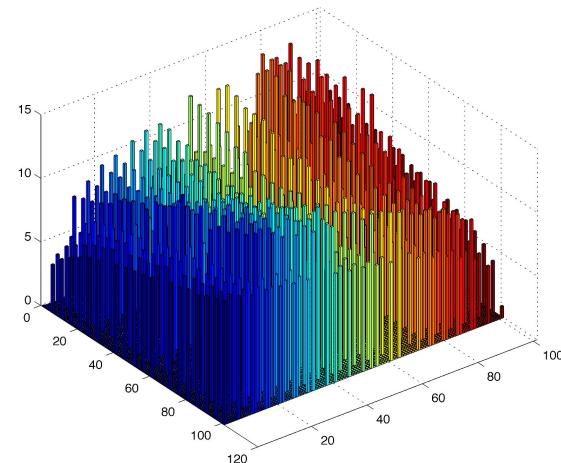
# Plan du cours

- Qu'est-ce que le traitement numérique des images
- L'image dans la machine
- Pourquoi a-t-on besoin de coder ?



# Qu'est-ce que le traitement numérique des images?

- Traitement d'images ou *image processing*
  - mathématiques appliqués à l'étude des images
  - informatique appliqué aux images
    - manipulation
    - analyse
    - interprétation



# Domaines d'application du traitement d'images



# Qui utilise le traitement d'images



PyTorch



TensorFlow



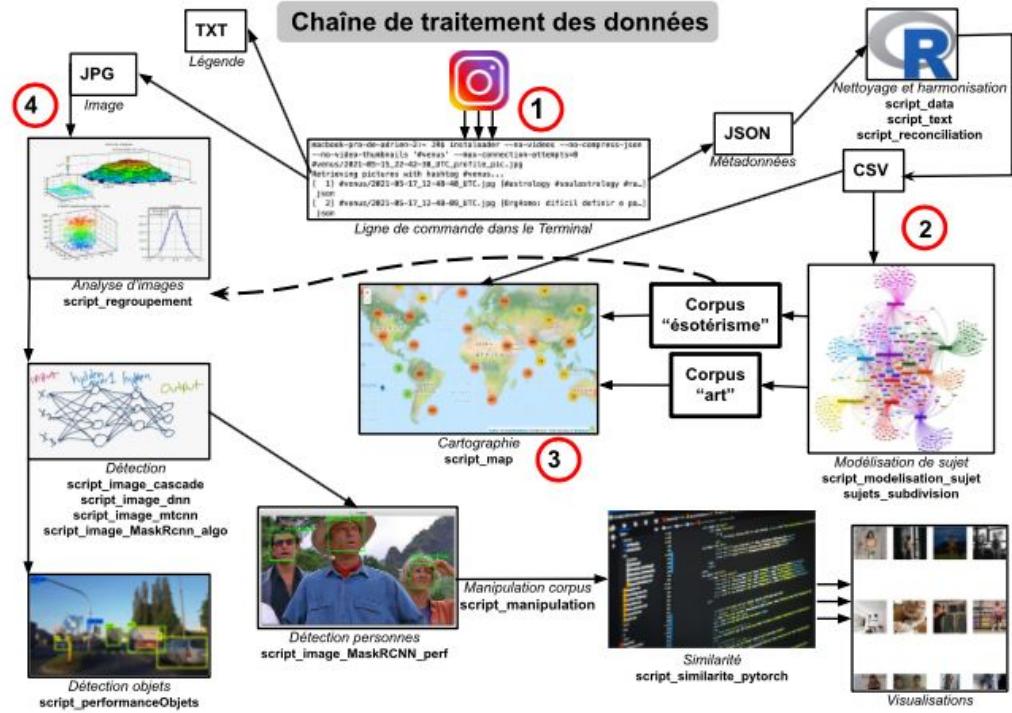
Keras



UNIVERSITÉ  
DE GENÈVE

# En quoi le traitement d'image m'est-il utile dans mes propres recherches ?

- Manipulation des images
  - Détection de personnes
  - Analyse de la composition
  - Similarité
- 
- Et vous ?



# L'image dans la machine

- Acquisition
- Stockage
- Bonnes pratiques
- Nature des images
- Formats



# Acquisition des images numériques

## Sources privées

- appareil photo numérique
- smartphone
- numérisation d'un document personnel
- etc...

### ▼ Plus d'infos :

Provenant de : iPhone  
Dimensions : 1328 × 1690

Marque du périphérique : Apple

Modèle du périphérique : iPhone 7

Espace colorimétrique : RGB

Profil de couleur : Display P3

Longueur focale : 3,99

Canal alpha : Non

Yeux rouges : Non

Type de mesure : 5

Numéro F : 1,8

Programme d'exposition : 2

Temps d'exposition : 1/17

### ▼ Plus d'infos :

Ouverture : dimanche, 17 janvier 2021 à 09:03

Dimensions : 6000 × 4000

Marque du périphérique : SONY

Modèle du périphérique : ILCE-6300

Espace colorimétrique : RGB

Profil de couleur : sRGB IEC61966-2.1

Longueur focale : 50

Canal alpha : Non

Yeux rouges : Non

Type de mesure : 5

Numéro F : 5,6

Programme d'exposition : 4

Temps d'exposition : 1/250



# Acquisition des images numériques

## Sources publiques

- collections muséales (base Joconde, MoMa)
- bibliothèques (BnF, Bodleian)
- centres iconographiques (Getty)
- sites spécialisés (IMDB, Internet Archive)
- etc...



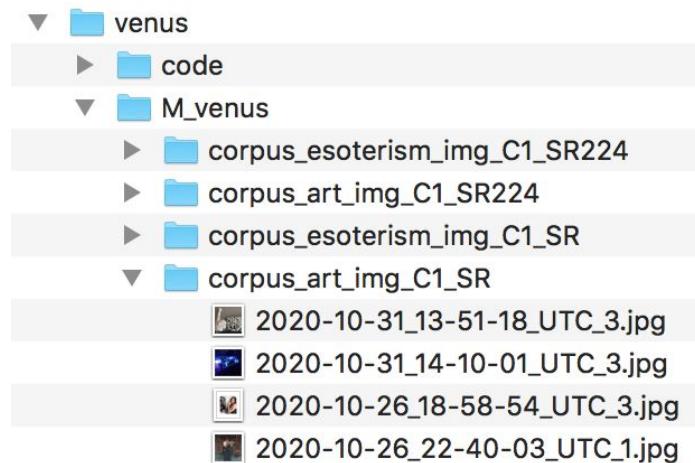
# Stockage des images

- Bonnes pratiques
- Natures
  - Images matricielles
  - Images vectorielles
- Formats
  - JPEG
  - TIFF
  - PNG
  - etc...



# Stockage des images : bonnes pratiques

- Organisation
- Nomenclature
- Homogénéité
- Réconciliation
- Réplicabilité



1994-06 [02]

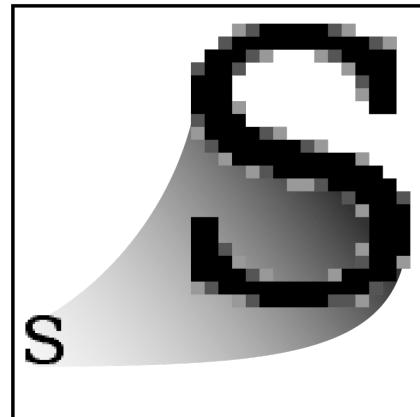


1994-06 [02]



# Stockage des images : natures

- Images matricielles
  - pixels (point) sous la forme d'une matrice
  - chaque point possède une ou plusieurs valeurs
  - photographie
- Images vectorielles
  - entités géométriques
  - formules mathématiques
  - dessins, graphiques



Raster  
.jpeg .gif .png

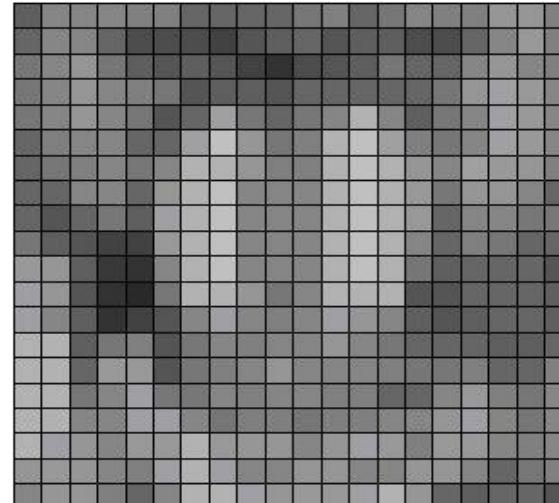
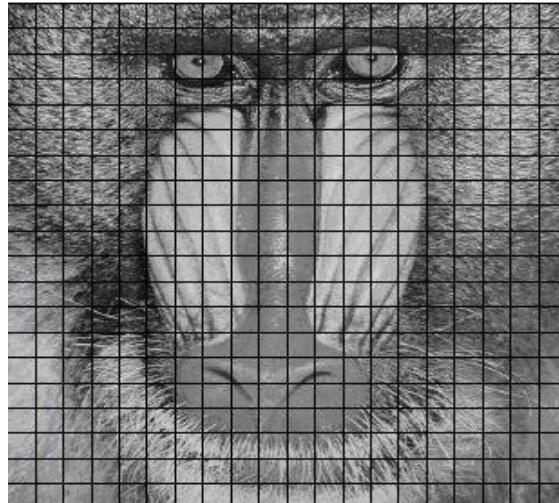


Vector  
.svg



# Stockage des images : images matricielles

- Du continu au discret
  - échantillonnage
  - quantification



# Stockage des images : images matricielles

- Du continu au discret
  - échantillonnage
  - quantification
- Représentation matricielle



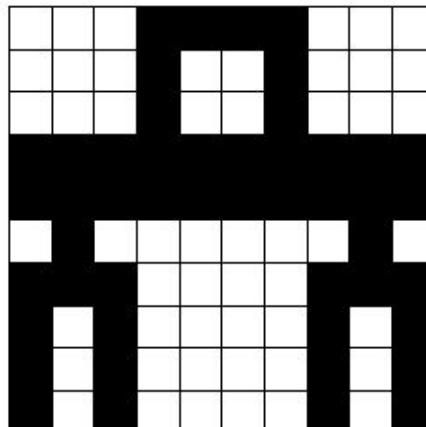
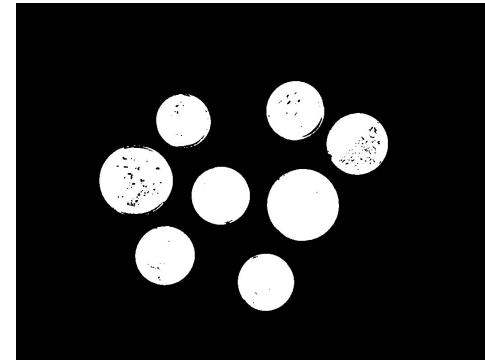
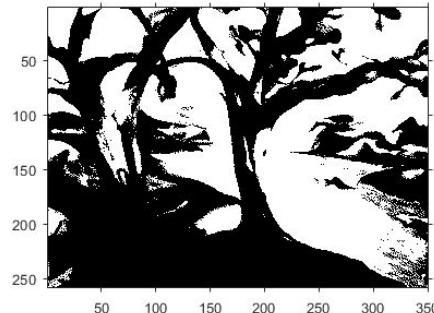
157	153	174	168	150	152	129	151	172	161	165	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	106	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	85	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	149	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	12	96	218

157	153	174	168	150	152	129	151	172	161	165	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	85	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	149	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218



# Stockage des images : images matricielles

- Du continu au discret
  - échantillonnage
  - quantification
- Représentation matricielle
- Caractéristiques principales
  - image binaire



→	0	0	0	1	1	1	1	0	0	0
→	0	0	0	1	0	0	1	0	0	0
→	0	0	0	1	0	0	1	0	0	0
→	1	1	1	1	1	1	1	1	1	1
→	1	1	1	1	1	1	1	1	1	1
→	0	1	0	0	0	0	0	0	1	0
→	1	1	1	0	0	0	0	1	1	1
→	1	0	1	0	0	0	0	1	0	1
→	1	0	1	0	0	0	0	1	0	1
→	1	0	1	0	0	0	0	1	0	1



# Stockage des images : images matricielles

- Du continu au discret
  - échantillonnage
  - quantification
- Représentation matricielle
- Caractéristiques principales
  - image binaire
  - image en niveau de gris



48	49	46	42	44
110	79	54	47	48
190	192	190	153	99
150	166	189	203	183
131	140	145	161	165

# Niveaux de gris

256



128



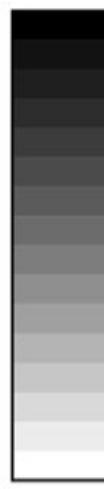
64



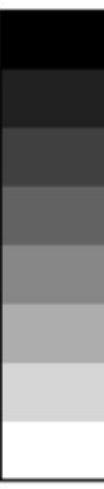
32



16



8



4



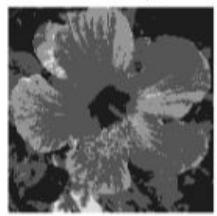
2



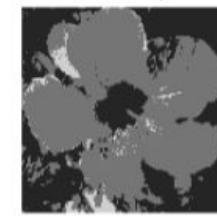
16 niveaux de gris



4 niveaux de gris



3 niveaux de gris



2 niveaux de gris



8 bits

7 bits

6 bits

5 bits

4 bits

3 bits

2 bits

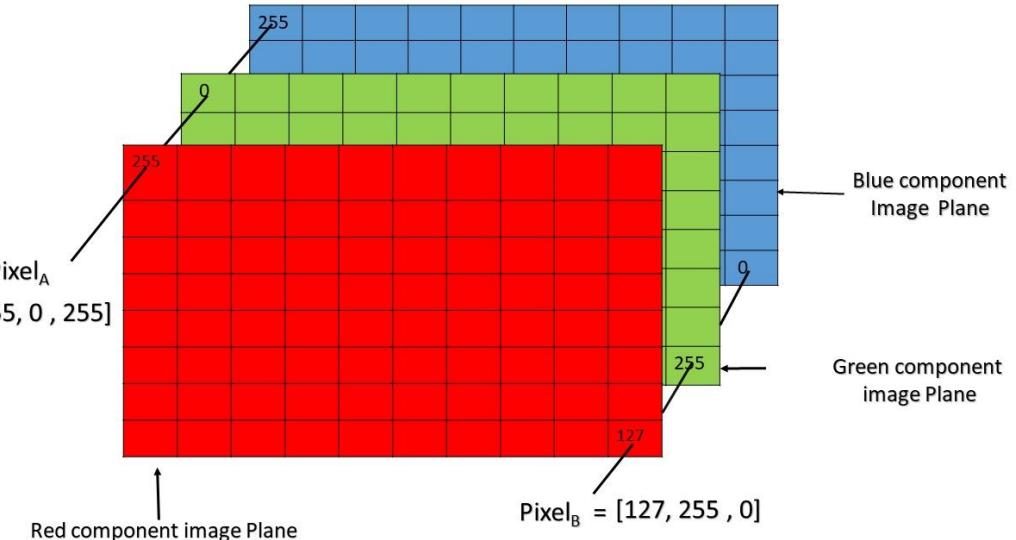
1 bit



UNIVERSITÉ  
DE GENÈVE

# Stockage des images : images matricielles

- Du continu au discret
  - échantillonnage
  - quantification
- Représentation matricielle
- Caractéristiques principales
  - image binaire
  - image en niveau de gris
  - image couleur



Pixel of an RGB image are formed from the corresponding pixel of the three component images



# *Formats*

# Stockage des images : formats

- **JPEG**
- **PNG**
- **GIF**
- **TIFF**
- **SVG**
- **EPS**
- **BMP**
- **XCF**
- **PSD**

# Stockage des images : formats compressés

- **JPEG (Joint Photographic Expert Group)**
  - réduction de la taille, perte d'information contenue : meilleur entre deux
- **PNG (Portable Network Graphic)**
  - compression sans perte, transparence, enregistre plusieurs millions de couleurs
- **GIF (Graphics Interchange Format)**
  - petite taille de fichier, enregistré sur 256 couleurs différentes, transparence, animation
- **TIFF (Tagged Image File Format)**
  - grande taille, sans perte d'information : pour l'impression (professionnel)

# Stockage des images : formats vectoriels

- **SVG (Scalable Vector Graphics)**
- **EPS (Encapsulated Post Script)**

# Stockage des images : formats bitmap

- **BMP (Bitmap)**
  - Développé par Microsoft pour Windows : format propriétaire
  - Aucune compression donc aucune perte
  - Grande taille
  - Pendant du format TIFF

# Stockage des images : formats multicouches

- **XCF (Gimp)**
- **PSD (Adobe, Photoshop)**

# Pourquoi ne pas utiliser Photoshop ?

- Comprendre ce qui se passe derrière l'interface
- Analyser ses images plus rapidement
- Homogénéisation et réplicabilité
- Pouvoir aller plus loin dans le traitement numérique des images



# Python

- Pourquoi Python ?
  - Facilité d'utilisation
  - Pour tous niveaux
  - Polyvalent
  - Courant
  - Développé



UNIVERSITÉ  
DE GENÈVE

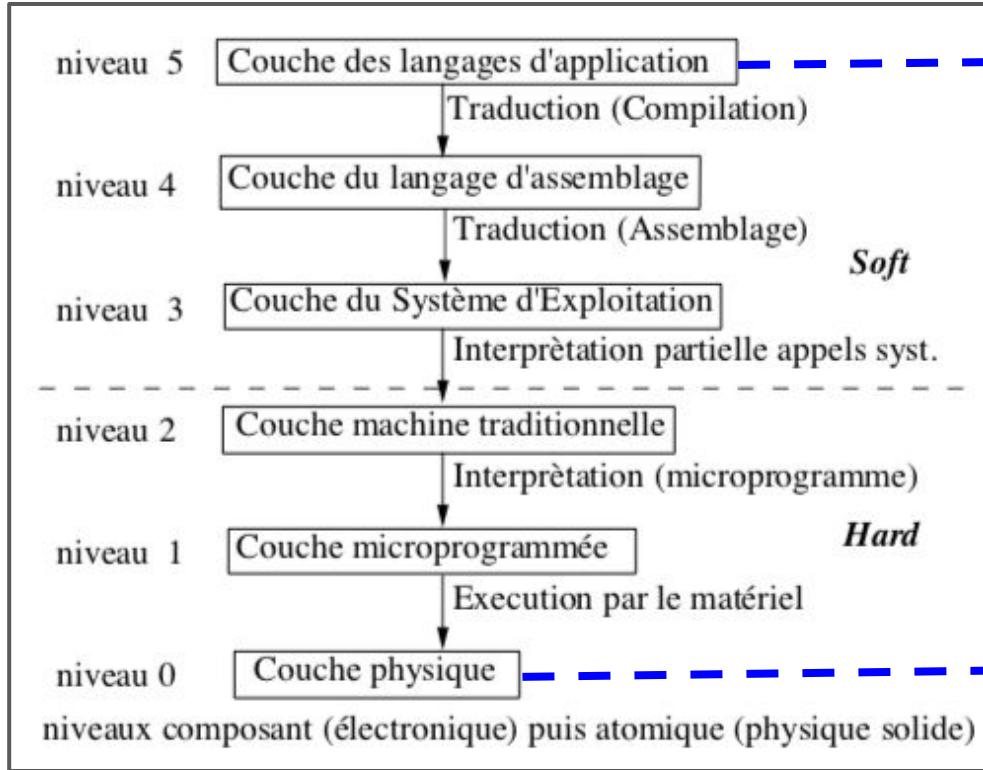


# Architecture des systèmes d'informatiques

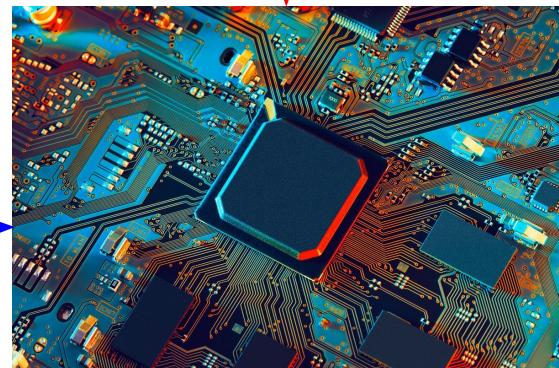
## Langages d'application

Top 10 Programming Languages		Python	C	Java	C++	C#	R	JavaScript	PHP	Go	Swift
Paradigm	Multi-paradigm: object-oriented, imperative, functional, procedural, reflective		Imperative (procedural), structured	Multi-paradigm: object-oriented (class-based), structured, imperative, generic, reflective, concurrent	Multi-paradigm: procedural, functional, object-oriented, generic	Multi-paradigm: structured, imperative, object-oriented, event-driven, task-driven, functional, generic, reflective, concurrent	Multi-paradigm: array, object-oriented, imperative, functional, procedural, generic, reflective, concurrent	Multi-paradigm: object-oriented (prototype-based), imperative, functional, procedural, event-driven	Imperative, object-oriented, procedural, reflective	Compiled, concurrent, imperative, structured	Multi-paradigm: protocol-oriented, object-oriented, functional, imperative, block-structured
Designed by	Guido van Rossum		Dennis Ritchie	James Gosling	Bjarne Stroustrup	Microsoft	Ross Ihaka and Robert Gentleman	Brendan Eich	Rasmus Lerdorf	Robert Griesemer, Rob Pike, Ken Thompson	Chris Lattner and Apple Inc
Developer	Python Software Foundation		Dennis Ritchie & Bell Labs (creators), ANSI X3J11 (ANSI C), ISO/IEC	Sun Microsystems (now owned by Oracle corporation)	Bell Labs	Microsoft	R Core Team	Netscape Communications Corporation, Mozilla Foundation, Ecma International	The PHP Development Team, Zend Technologies	Google Inc.	Apple Inc
First appeared	20 February 1991 (26 years ago)		1972 (45 years ago)	May 23 1995 (22 years ago)	1983 (34 years ago)	2000 (17 years ago)	August 1993 (24 years ago)	December 4, 1995 (21 years ago)	June 8, 1995 (22 years ago)	November 10, 2009 (7 years ago)	June 2, 2014 (3 years ago)
Typing discipline	Duck, dynamic, strong		Static, weak, manifest, nominal	Static, strong, safe, nominative, manifest	Static, nominative, partially inferred	Static, dynamic, strong, safe, nominative, partially inferred	Dynamic	Dynamic, duck	Dynamic, weak, gradual (as for PHP 7.0.0)	Strong, static, inferred, structural	Static, strong, inferred
Platform	Cross-platform		Cross-platform	Windows, Solaris, Linux, OS X	Linux, MacOS, Solaris	Common Language Infrastructure	UNIX platforms, Windows, MacOS	Cross-platform	Unix-like, Windows	Linux, macOS, FreeBSD, NetBSD, OpenBSD, Windows, Plan 9, DragonFly BSD, Solaris	Darwin, Linux, FreeBSD
Filename extensions	.py, .pyc, .pyo (prior to 3.5), .pyw, .pyz (since 3.5)		.c, .h	.java, .class, .jar	.cc, .cpp, .C, c++, .h, .hh, .hpp, .hxx, .h++	.cs	.r, .R, .RData, .rds, .rda	.js	.php, .phml, .php3, .php4, .php5, .php7, .phps	.go	.swift

# Architecture des systèmes d'informatiques



```
31 self._fp = None
32 self.fingerprints = set()
33 self.logdups = True
34 self.debug = debug
35 self.logger = logging.getLogger(__name__)
36 if path:
37     self._fp = open(os.path.join(path, 'fingerprint'))
38     self._fp.seek(0)
39     self.fingerprints.update(self._fp.readlines())
40
41 @classmethod
42 def from_settings(cls, settings):
43     debug = settings.getboolean('REPORTING_DEBUG')
44     return cls(job_dir(settings), debug)
45
46 def request_seen(self, request):
47     fp = self.request_fingerprint(request)
48     if fp in self.fingerprints:
49         return True
50     self.fingerprints.add(fp)
51     if self._fp:
52         self._fp.write(fp + os.linesep)
53
54     def request_fingerprint(self, request):
55         return request_fingerprint(request)
56
```



# Python

- Pourquoi Python ?
- Quelques liens utiles
  - Tutoriels
    - <https://www.w3schools.com/python/>
    - <https://openclassrooms.com/en/courses/7168871-apprenez-les-bases-du-langage-python>
    - <https://docs.python.org/fr/3/tutorial/>



## Objectifs pédagogiques

À la fin de ce cours, vous serez capable de :

- Créer des données avec Python.
- Gérer la logique de programmation dans Python.
- Extraire des données du web avec Python et certains de ses packages.

### Learning by Examples

With our "Try it Yourself" editor, you can edit Python code and view the result.

#### Example

```
print("Hello, World!")
```

Try it Yourself >

- 1. Mise en bouche
- 2. Mode d'emploi de l'interpréteur Python
  - 2.1. Lancement de l'interpréteur
    - 2.1.1. Passage d'arguments
    - 2.1.2. Mode interactif
  - 2.2. L'interpréteur et son environnement
    - 2.2.1. Encodage du code source
- 3. Introduction informelle à Python
  - 3.1. Utilisation de Python comme une calculatrice
    - 3.1.1. Les nombres
    - 3.1.2. Chaînes de caractères
    - 3.1.3. Listes
  - 3.2. Premiers pas vers la programmation



# Python

- Pourquoi Python ?
- Quelques liens utiles

- Tutoriels
- Conventions
  - PEP 8 (<https://pep8.org>)
    - variables : lettres, underscore (image, image\_gris, image\_gris\_ballon, etc.)
    - classes : lettre, majuscules (Filtre, FiltreGris, etc..)
    - mots-clés réservés (keywords)
    - # pour les commentaires
    - code couleur

and	exec	not
assert	finally	or
break	for	pass
class	from	print
continue	global	raise
def	if	return
del	import	try
elif	in	while
else	is	with
except	lambda	yield

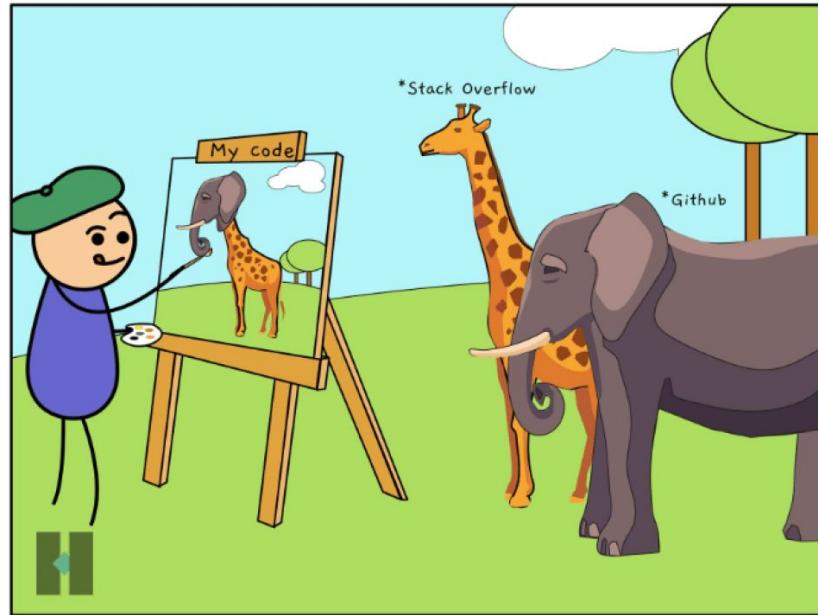


```
1 import unittest
2 import quotes
3
4 class MyTests (unittest.TestCase):
5
6     def test_add_get_quote (self):
7         quotes.add ("Confucius ", "A journey of a thousand miles ... ")
8         q = quotes.get ("Confucius ", contains ="step")
9         self.assertEqual (q, [ "A journey of a thousand miles ... "])
10
11    def test_add_get_quote_no_contains (self):
12        quotes.add ("Confucius ", "A journey of a thousand miles ... ")
13        q = quotes.get ("Confucius ")
14        self.assertEqual (q, [ "A journey of a thousand miles ... "])
15
16 if __name__ == "__main__":
17     unittest.main()
```



# Python

- Pourquoi Python ?
- Quelques liens utiles
  - Tutoriels
  - Conventions
  - Aide
    - ?fonction
    - <https://docs.python.org/3/>
    - <https://stackoverflow.com>
    - <https://www.geeksforgeeks.org>
    - <https://github.com>
      - Astuces
        - en anglais, par mots-clés ou sous forme de question
        - copier un bout de code
        - regarder plusieurs solutions



# Outils en Python pour les images

## Libraires communes

- numpy
- os
- matplotlib

## Libraires spécialisées

- opencv
- pillow (PIL)
- scikit-image



# Pour la semaine prochaine

Exercez-vous à Python !

BEFORE AND AFTER CODING



c++



javascript



java



python



UNIVERSITÉ  
DE GENÈVE