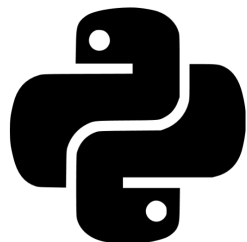


ThoughtWorks®

não precisa importar

PYTHON BUILTINS

Um passeio pelas funções e classes embutidas no Python 3.8.



JUST PYTHON
release v3.0
2019-11-09

Luciano Ramalho
@ramalhoorg

ThoughtWorks®

HELP

É sempre bom poder pedir ajuda

PRIMEIRA OLHADA: DIR E HELP

```
Python 3.8.0 (v3.8.0:fa919fdf25, Oct 14 2019, 10:23:27)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
```

```
>>> dir()
['__annotations__', '__builtins__', '__doc__', '__loader__', '__name__', '__package__',
 '__spec__']
>>> help(__builtins__)
```

Help on built-in module builtins:

NAME

builtins – Built-in functions, exceptions, and other objects.

DESCRIPTION

Noteworthy: None is the `nil' object; Ellipsis represents `...' in slices.

CLASSES

object

 BaseException

 Exception

 ArithmeticError

 FloatingPointError

 OverflowError

 ZeroDivisionError

 AssertionError

 AttributeError

 BufferError

**8618 linhas
de ajuda**

LISTA HIERÁRQUICA DE CLASSES NO HELP

```
UnicodeEncodeError
UnicodeTranslateError
Warning
    BytesWarning
    DeprecationWarning
    FutureWarning
    ImportWarning
    PendingDeprecationWarning
    ResourceWarning
    RuntimeWarning
    SyntaxWarning
    UnicodeWarning
    UserWarning
GeneratorExit
KeyboardInterrupt
SystemExit
bytearray
bytes
classmethod
complex
dict
enumerate
filter
float
frozenset
int
    bool
list
```

Um monte de
exceções...

Classes com
nomes em caixa
baixa



bool é
subclasse de
int!



USANDO PYTHON PARA APRENDER PYTHON

Como é bom ter um console interativo

LISTA DE NOMES EM `__BUILTINS__`

```
>>> print(*dir(__builtins__), sep=' ')
ArithmeticError AssertionError AttributeError BaseException BlockingIOError
BrokenPipeError BufferError BytesWarning ChildProcessError
ConnectionAbortedError ConnectionError ConnectionRefusedError
ConnectionResetError DeprecationWarning EOFError Ellipsis EnvironmentError
Exception False FileNotFoundError FloatingPointError
FutureWarning GeneratorExit IOError ImportError ImportWarning
IndentationError IndexError InterruptedError IsADirectoryError KeyError
KeyboardInterrupt LookupError MemoryError ModuleNotFoundError NameError None
NotADirectoryError NotImplemented NotImplementedError OSError OverflowError
PendingDeprecationWarning PermissionError ProcessLookupError RecursionError
ReferenceError ResourceWarning RuntimeError RuntimeWarning StopAsyncIteration
StopIteration SyntaxError SyntaxWarning SystemError SystemExit TabError
TimeoutError True TypeError UnboundLocalError UnicodeDecodeError
UnicodeEncodeError UnicodeError UnicodeTranslateError UnicodeWarning
UserWarning ValueError Warning ZeroDivisionError _ __build_class__ __debug__
__doc__ __import__ __loader__ __name__ __package__ __spec__ abs all any
ascii bin bool breakpoint bytearray bytes callable chr classmethod
compile complex copyright credits delattr dict dir divmod enumerate eval
exec exit filter float format frozenset getattr globals hasattr hash
help hex id input int isinstance issubclass iter len license list
locals map max memoryview min next object oct open ord pow print
property quit range repr reversed round set setattr slice sorted
staticmethod str sum super tuple type vars zip
```

ÁRVORE DE EXCEÇÕES

Ou: como listar uma hierarquia de classes

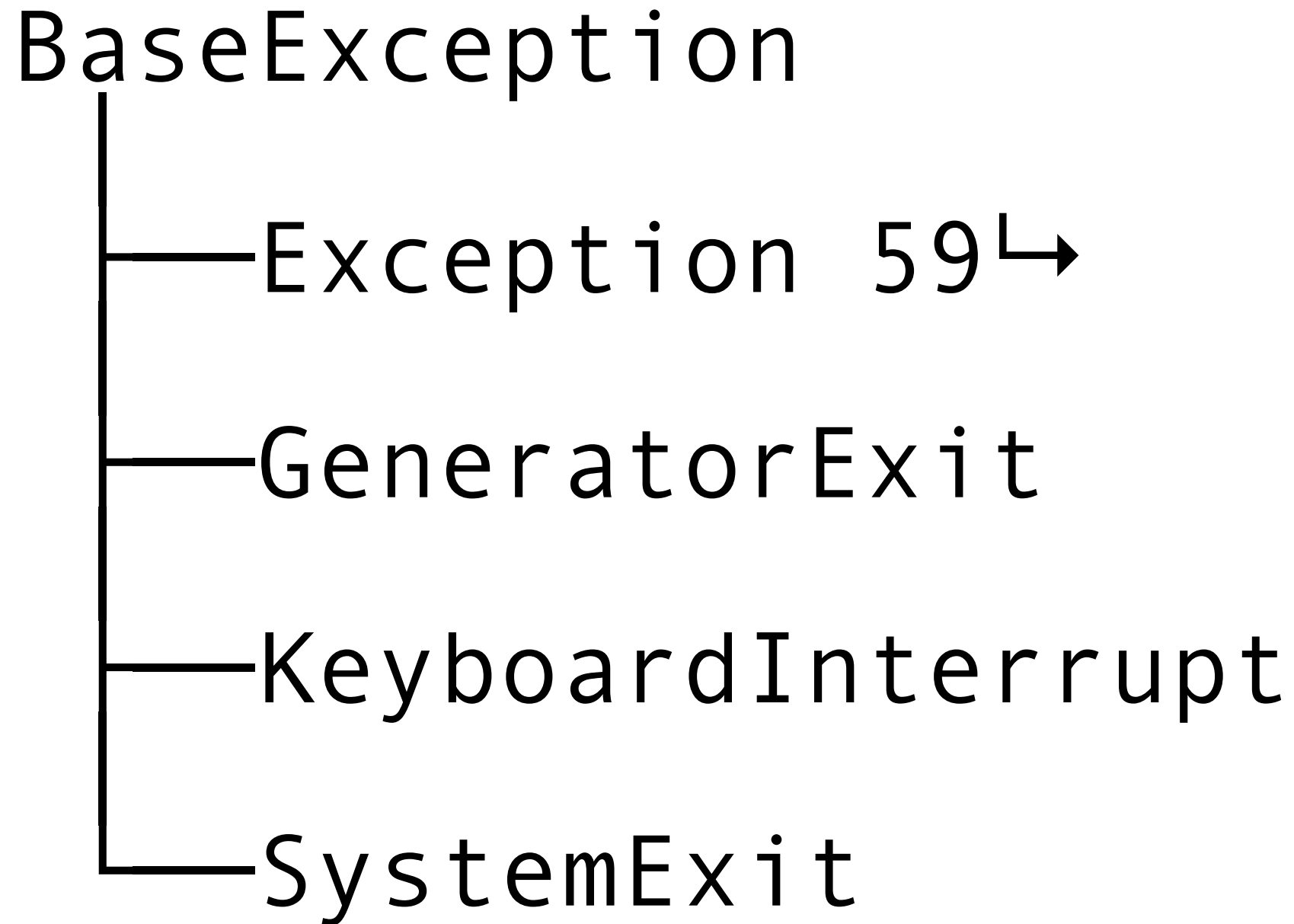
LISTA DE NOMES EM CAIXA-MISTA, SÓ INVOCÁVEIS

```
>>> Names = [s for s in dir(__builtins__) if s.lower() != s]
>>> Exceptions = [s for s in Names if callable(getattr(__builtins__, s))]
>>> print(*Exceptions, sep=' ')
ArithmeticError  AssertionError  AttributeError  BaseException
BlockingIOError  BrokenPipeError  BufferError  BytesWarning
ChildProcessError  ConnectionAbortedError  ConnectionError
ConnectionRefusedError  ConnectionResetError  DeprecationWarning  EOFError
EnvironmentError  Exception  FileExistsError  FileNotFoundError
FloatingPointError  FutureWarning  GeneratorExit  IOError  ImportError
ImportWarning  IndentationError  IndexError  InterruptedError
IsADirectoryError  KeyError  KeyboardInterrupt  LookupError  MemoryError
ModuleNotFoundError  NameError  NotADirectoryError  NotImplementedError
OSError  OverflowError  PendingDeprecationWarning  PermissionError
ProcessLookupError  RecursionError  ReferenceError  ResourceWarning
RuntimeError  RuntimeWarning  StopAsyncIteration  StopIteration
SyntaxError  SyntaxWarning  SystemError  SystemExit  TabError
TimeoutError  TypeError  UnboundLocalError  UnicodeDecodeError
UnicodeEncodeError  UnicodeError  UnicodeTranslateError  UnicodeWarning
UserWarning  ValueError  Warning  ZeroDivisionError
```

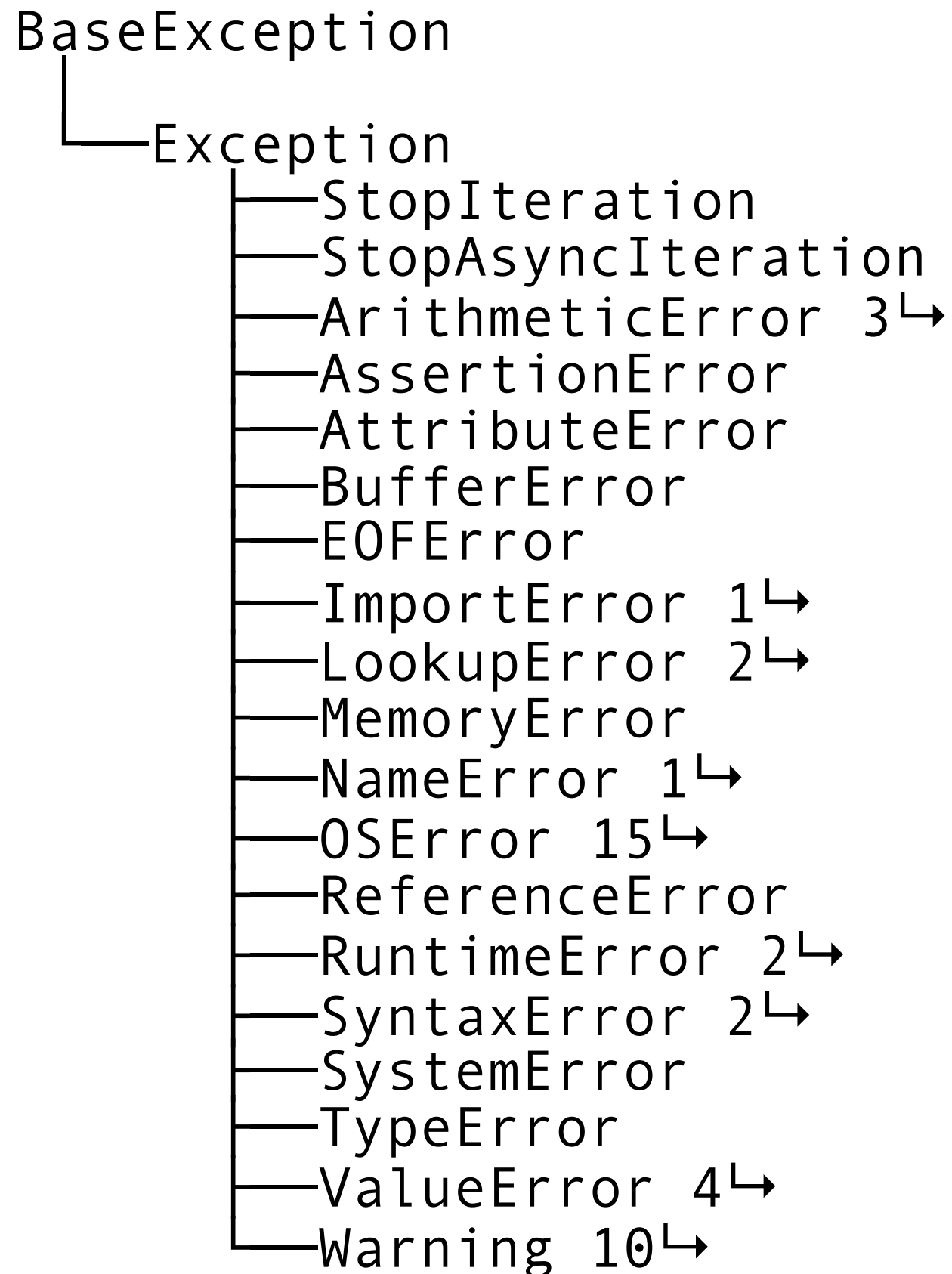

HIERARQUIA DE EXCEÇÕES

```
>>> def tree(cls, level=0):
...     yield ' ' * 4 * level + cls.__name__
...     for sub in sorted(cls.__subclasses__(), key=lambda c: c.__name__):
...         yield from tree(sub, level+1)
...
>>> print(*tree(BaseException), sep='\n')
BaseException
    Exception
        ArithmeticError
            FloatingPointError
            OverflowError
            ZeroDivisionError
        AssertionError
        AttributeError
        BufferError
        EOFError
        EndOfBlock
        Error
        ErrorDuringImport
        ImportError
            ModuleNotFoundError
            ZipImportError
        LookupError
            CodecRegistryError
            IndexError
```

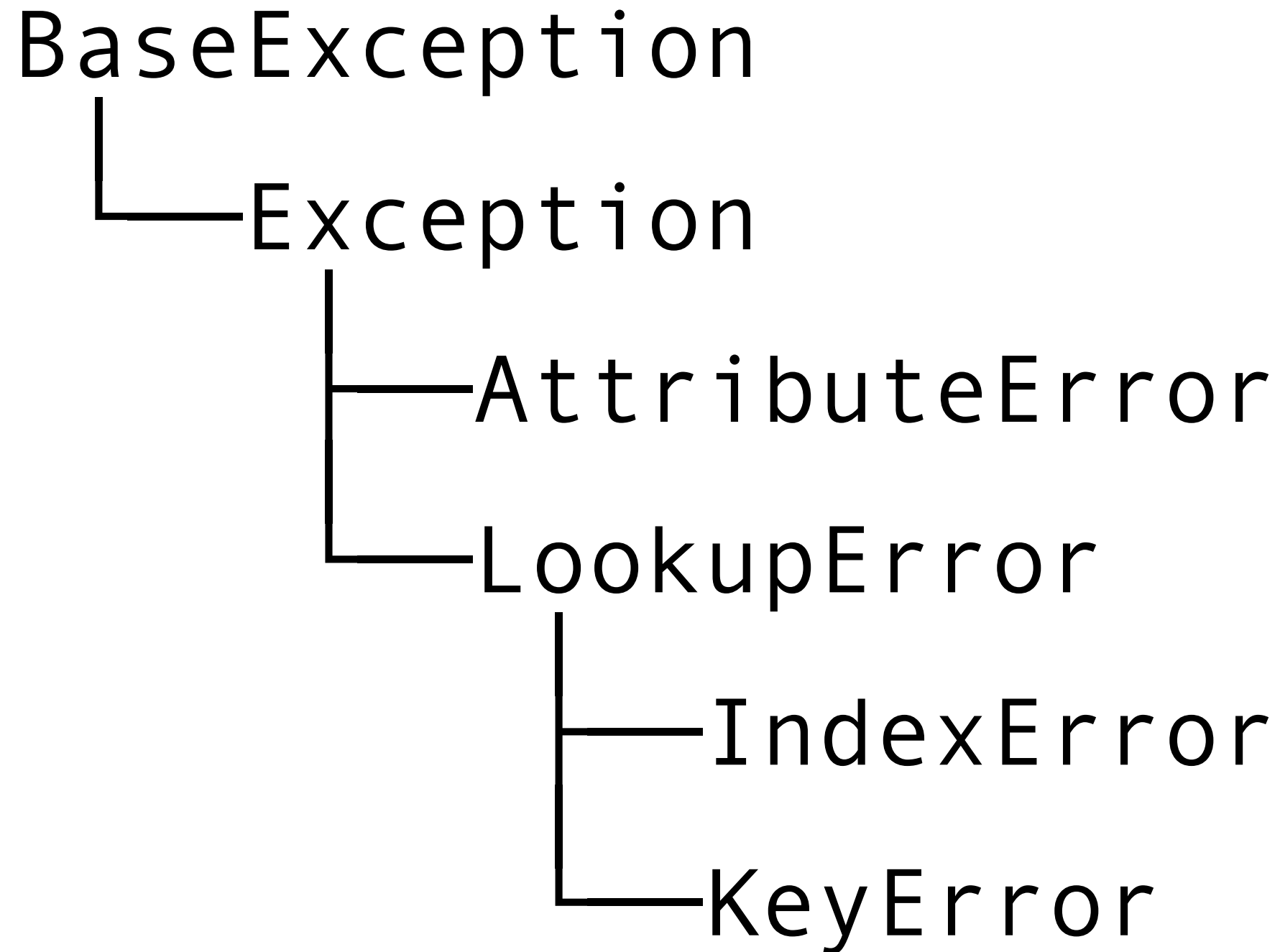
TOPO DA HIERARQUIA DE EXCEÇÕES



SUBCLASSES DIRETAS DE EXCEPTION



ERROS DE ACESSO A ATRIBUTOS OU ITENS



ERROS DE UNICODE

BaseException

└─ Exception

└─ ValueError

└─ UnicodeError

└─ UnicodeDecodeError

└─ UnicodeEncodeError

└─ UnicodeTranslateError

OUTROS EMBUTIDOS

A melhor parte dos built-ins: funções e classes

LISTA DE NOMES EM CAIXA-BAIXA SEM PREFIXO _

```
>>> names = [s for s in dir(__builtins__) if s == s.lower() and s[0] != '_']
>>> print(*names, sep=' ')
abs all any ascii bin bool breakpoint bytearray bytes callable chr
classmethod compile complex copyright credits delattr dict dir divmod
enumerate eval exec exit filter float format frozenset getattr
globals hasattr hash help hex id input int isinstance issubclass
iter len license list locals map max memoryview min next object oct
open ord pow print property quit range repr reversed round set
setattr slice sorted staticmethod str sum super tuple type vars zip
```

MESMA LISTA DE ANTES, AGORA COM TIPOS

```
>>> categorized = []
>>> for name in names:
...     obj = getattr(__builtins__, name)
...     categorized.append((type(obj).__name__, name))
...
>>> print(*categorized, sep='\n')
('builtin_function_or_method', 'abs')
('builtin_function_or_method', 'all')
('builtin_function_or_method', 'any')
('builtin_function_or_method', 'ascii')
('builtin_function_or_method', 'bin')
('type', 'bool')
('builtin_function_or_method', 'breakpoint')
('type', 'bytearray')
('type', 'bytes')
('builtin_function_or_method', 'callable')
('builtin_function_or_method', 'chr')
('type', 'classmethod')
('builtin_function_or_method', 'compile')
('type', 'complex')
('_Printer', 'copyright')
('_Printer', 'credits')
('builtin_function_or_method', 'delattr')
('type', 'dict')
('builtin_function_or_method', 'dir')
```


MESMA LISTA DE ANTES, AGRUPADA POR TIPOS

```
>>> for category, group in groupby(sorted(categorized), itemgetter(0)):
...     print('-' * 20, category)
...     for _, name in group:
...         print(name)
...
_____ Quitter
exit
quit
_____ _Helper
help
_____ _Printer
copyright
credits
license
_____ builtin_function_or_method
abs
all
any
ascii
bin
breakpoint
callable
chr
compile
delattr
```

FUNÇÕES

| | | |
|------------|------------|---------|
| abs | format | min |
| all | getattr | next |
| any | globals | oct |
| ascii | hasattr | open |
| bin | hash | ord |
| breakpoint | hex | pow |
| callable | id | print |
| chr | input | repr |
| compile | isinstance | round |
| delattr | issubclass | setattr |
| dir | iter | sorted |
| divmod | len | sum |
| eval | locals | vars |
| exec | max | |

CLASSES

| | |
|-------------|--------------|
| bool | object |
| bytearray | property |
| bytes | range |
| classmethod | reversed |
| complex | set |
| dict | slice |
| enumerate | staticmethod |
| filter | str |
| float | super |
| frozenset | tuple |
| int | type |
| list | zip |
| map | |
| memoryview | |

LISTA DE NOMES EM CAIXA-MISTA, NÃO INVOCÁVEIS

```
>>> Const = [s for s in Names if not callable(getattr(__builtins__, s))]  
>>> print(*Const, sep=' ')  
Ellipsis False None NotImplemented True
```

LISTA DE NOMES EM CAIXA-MISTA, NÃO INVOCÁVEIS

```
>>> Const = [s for s in Names if not callable(getattr(__builtins__, s))]  
>>> print(*Const, sep=' ')  
Ellipsis False None NotImplemented True
```

Conferindo que os nomes acima são todos os nomes em caixa-mista que não são classes de exceções:

```
>>> set(Names) - set(Exceptions)  
{'Ellipsis', 'NotImplemented', 'False', 'None', 'True'}
```

INVOCÁVEIS BUILT-IN SEPARADOS EM CATEGORIAS

| FUNÇÕES BÁSICAS | TIPOS DE DADOS | PROCESSAMENTO | FERRAMENTAS DE PROGRAMAÇÃO | |
|---|--|---|---|--|
| exibição de objetos ascii format repr str* | tipos simples bool complex float int object | operações com coleções all any len max min sorted sum | decoradores de métodos classmethod property staticmethod | operações com atributos delattr getattr hasattr setattr |
| exibição de inteiros bin hex oct | sequências planas bytearray bytes memoryview str* | iteradores/geradores enumerate filter map range reversed zip | delegação dinâmica super | inspeção do ambiente breakpoint dir* globals locals vars |
| entrada/saída input open print | containers dict frozenset list set tuple | iteração em baixo nível iter next | introspecção de objetos callable dir* hash id isinstance issubclass type* | utilitárias compile eval exec slice type* __import__ |
| unicode chr ord | | | singletons especiais Ellipsis False None NotImplemented True | |
| matemática abs divmod pow round | | | | |

ANOTHER WAY TO DO A BIG TITLE OR SECTION DIVIDER

Add a subhead if you want.



ThoughtWorks®

FULL BLEED PHOTO WITH TEXT OVERLAY

Place the text wherever the photo subjects dictates. You can delete this subhead and the keyline above.

YOUR BASIC, HARD-WORKING CONTENT SLIDE

A content headline

Did you know hitting tab at the beginning of a line demotes it from a headline (default) to body copy (what you see here)?

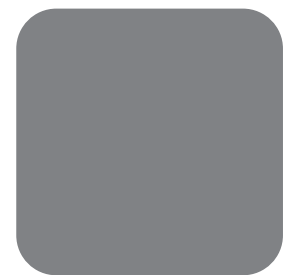
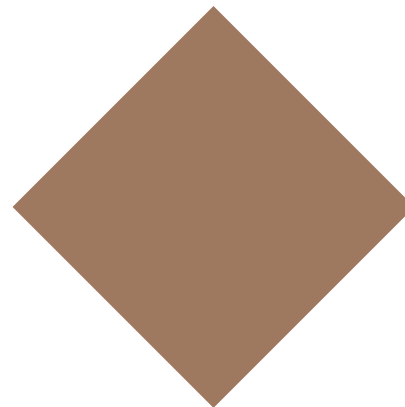
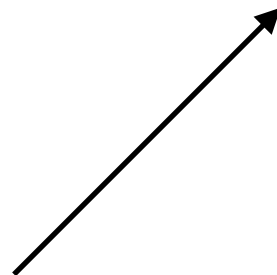
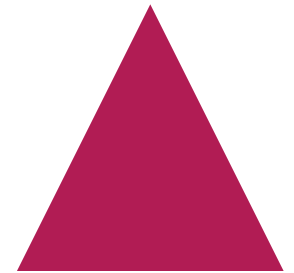
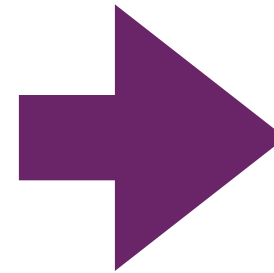
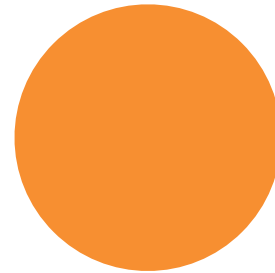
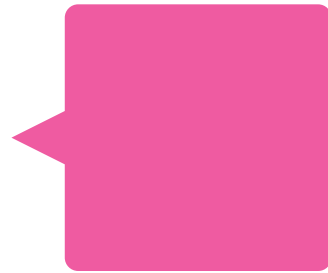
Shift-tab makes a headline again.

And back down to body by hitting tab again.

- Tab one more time to start bullets
- Enter begins the next bullet
 - Tab again for sub-bullets
 - And again for tiny bullets

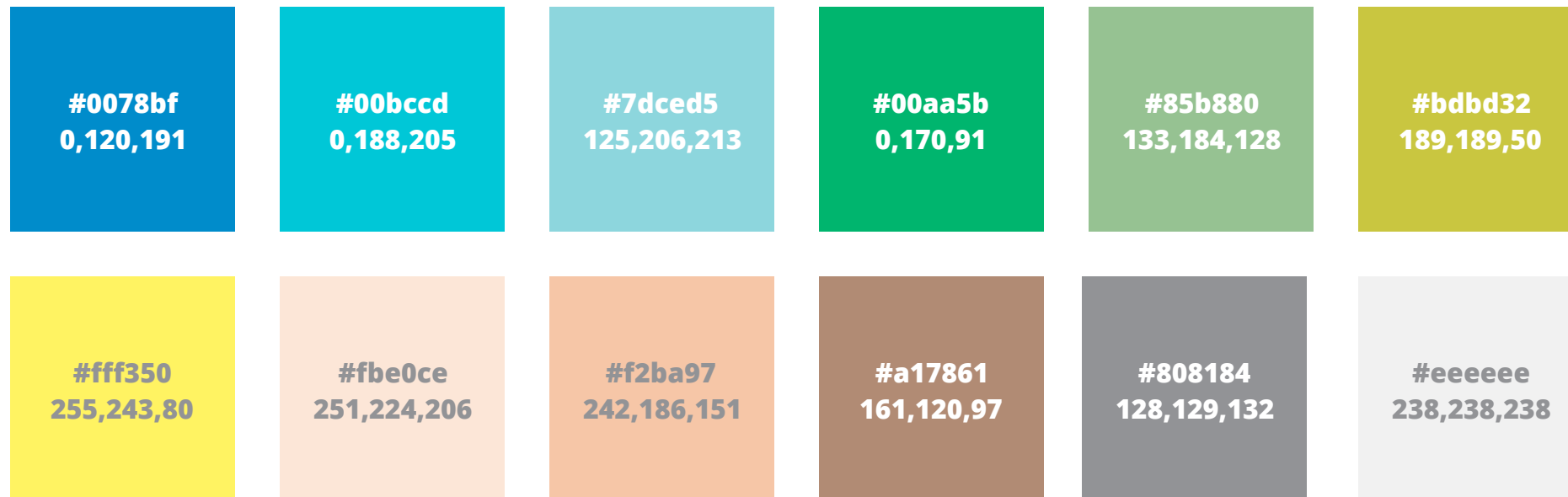
Sometimes you need a blank template.

SHAPES

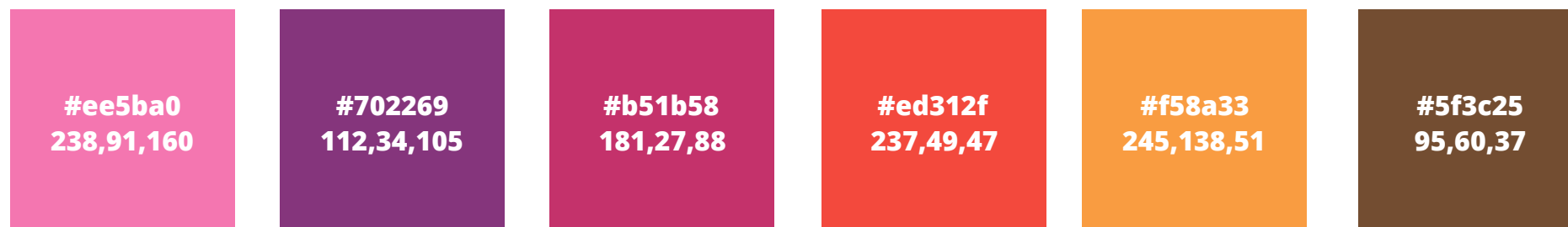


COLOR-PALETTE

Safe on Projectors



Risky on Projectors



BOXES, SHAPES AND TABLES

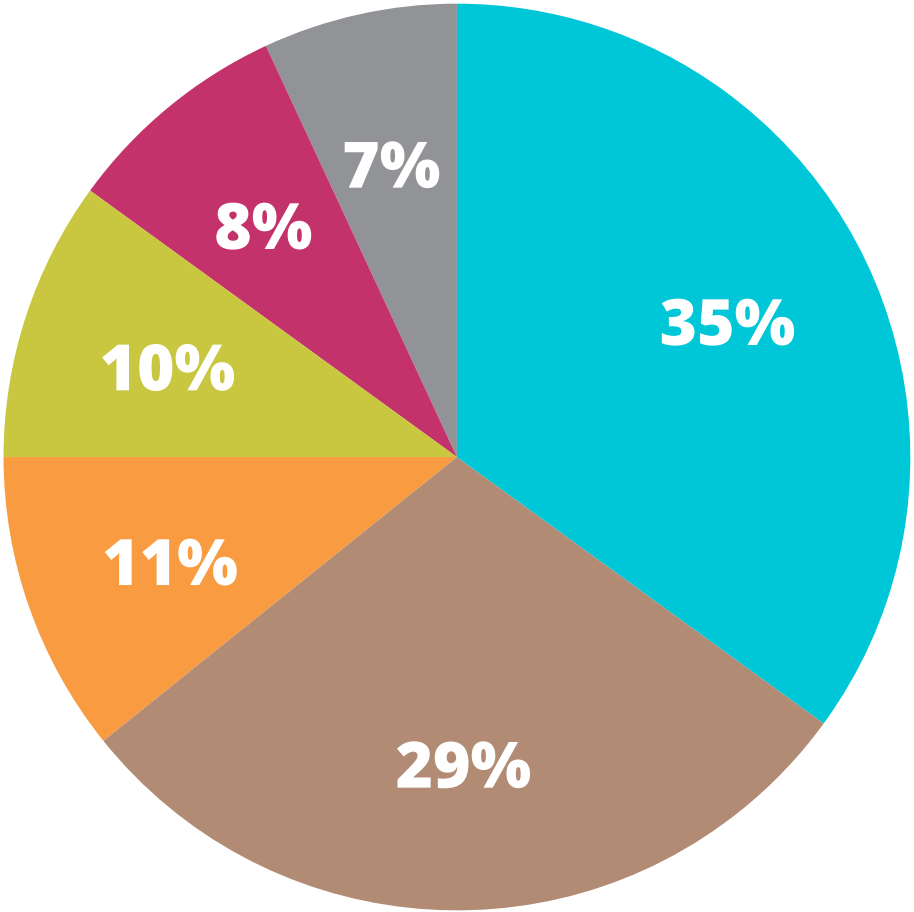
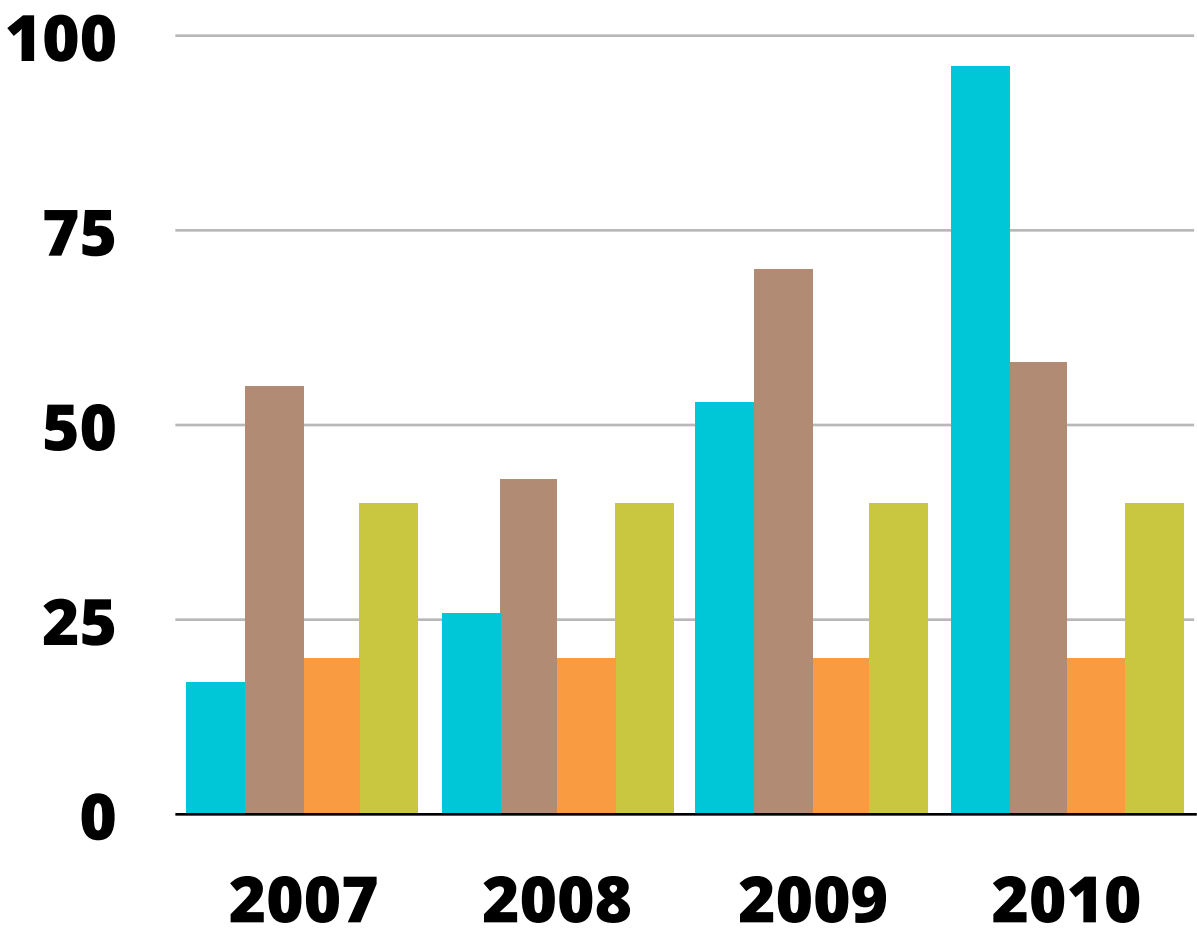
By default, a floating text box looks like this

Shapes are tinted gray, with lots of padding.

Works for any shape

| | | |
|--------------------|---------------------------|-----------------------------|
| | Header rows ... | are black. |
| Header columns ... | If you insert a table ... | it should automatically ... |
| are gray. | pick up this style. | Enjoy. |

CHARTS



PARTING THOUGHTS (USING THE CHECKLIST TEMPLATE)

First, get the proper font: <http://opensans.com/>

Start with a clean template.

When in doubt, copy and paste from something that already works.

Be brave:

- Learn to use master pages properly.
- Edit! Brevity, not design, makes presentations great.

DON'T MISS MASTER SLIDES IN OTHER COLORS

- Black
- Blue
- Khaki
- Magenta
- Orange
- Green
- Pink

THANK YOU

For questions or suggestions:

Go to Brand Hub

<https://my.thoughtworks.com/groups/brand>

ThoughtWorks®