
SuperPython Documentation

Versão 1.1.0

Carlo Oliveira

20 October, 2015

1	SuperPython - Introdução	3
2	SuperPython - Módulos	5
3	Notas de Lançamento V. 0.1.0	7
3.1	Milestone	7
3.2	Aspectos do Lançamento	7
3.3	Melhoramentos	7
3.4	Consertos	8
3.5	Questões e Problemas Conhecidos	8
3.6	Lançamentos Anteriores e Posteriores	8
4	SuperPython - Módulos Principais	9
4.1	SuperPython - Pacote Cliente	9
5	SuperPython - Módulos Principais	11
5.1	SuperPython - Teste	11
6	Indices and tables	13
	Índice de Módulos do Python	15

Contents:

SuperPython - Introdução

SuperPython - Modulos

SuperPython é programado em [Brython](#)

Funcionalidades Documentadas:

- Modelo do SuperPython : Entidades Basicas *SuperPython - Módulos Principais*
- Testes Unitários do SuperPython : *SuperPython - Módulos Principais*

Notas de Lançamento V. 0.1.0

SuperPython

3.1 Milestone

Bruce - Protótipo da interface Gráfica

3.2 Aspectos do Lançamento

3.2.1 Destaques dos Aspectos

Este ambiente inicial permite apenas o teste da interface com o usuário

3.2.2 Aspecto #1

O ambiente apenas apresenta o visual da interface mas não permite nenhuma interação.

3.3 Melhoramentos

Versão para uso do Google Application Engine e PyBuilder.

3.3.1 Melhoramento #1

Foi adicionada uma configuração para uso do [Google_Cloud](#). A configuração *app.yaml* define os aspectos necessários para rodar no servidor do Google Application Engine.

3.3.2 Melhoramento #2

Foi adicionada uma configuração para uso do construtor de aplicativos [PyBuilder](#). A configuração *build.py* define os aspectos necessários para verificar testes, cobertura, cabeçalhos e detalhes para deployment.

3.4 Consertos

Nenhum conserto notável.

3.5 Questões e Problemas Conhecidos

A funcionalidade ainda é muito simples, requer melhorias.

Uma nova versão deve suportar o monitoramento da atividade dos alunos.

3.6 Lançamentos Anteriores e Posteriores

Próximo Lançamento: A ser definido *Lançamento 0.2.0*

SuperPython - Módulos Principais

4.1 SuperPython - Pacote Cliente

Adiciona um editor Ace, dois botões e dois consoles do programa.

class `client.superpython.core.Ace` (*browser, edit, project, code*)
Inclui uma janela com um editor Acejs.

Parâmetros

- **browser** – Brython module browser
- **edit** – Referência ao módulo editor Ace
- **project** – Projeto que o usuário está desenvolvendo
- **code** – Texto do código a ser adicionado no editor

add_editor (*code=None*)

annotate (*row=1, message='indefinido'*)

get_content ()

set_content (*code*)

test_dirty (_, *code_saved=False*)

Confere e testa o estado de edição para detectar modificações.

:returns Se o código foi modificado desde a última vez que foi salvo.

class `client.superpython.core.Console` (*browser, ace*)
Classe que define o console de resposta da execução

Parâmetros

- **browser** – Referência ao módulo navegador do Brython
- **ace** – Referência ao módulo editor Ace

beforerun ()

display_canvas (*display='block'*)

display_saved (*message='SAVED'*)

onexec_error ()

write (*data*)

class `client.superpython.core.Dims` (*x, y, w, h*)

Bases: `tuple`

`Dims.h`
Alias for field number 3

`Dims.w`
Alias for field number 2

`Dims.x`
Alias for field number 0

`Dims.y`
Alias for field number 1

class `client.superpython.core.SuperPython` (*browser, edit, project*)

Classe que define o ambiente de desenvolvimento

Parâmetros **browser** – Referência ao módulo navegador do Brython

load (*_=0, msg=None*)

logout_on_exit (*ev*)

main (*name='', code='# main'*)

save (*_=0, autosaved=False*)

Veja também:

Module `client.superpython.core`

Nota: Unidade de Modelo Cliente.

SuperPython - Módulos Principais

5.1 SuperPython - Teste

Verifica a funcionalidade do cliente web.

```
class superpython_tests.SuperPythonTest (methodName='runTest')
```

```
Bases: unittest.case.TestCase
```

```
    SuperPythonTest.setUp()
```

```
    SuperPythonTest.test_main()
```

```
        garante que intância de SuperPython é criada.
```

Veja também:

Modules *SuperPython - Módulos Principais*

Nota: Módulo principal de testes unitários.

Indices and tables

- `genindex`
- `modindex`
- `search`

C

`client.superpython.core` (*Web*), 9

S

`superpython_tests` (*Web*), 11

A

Ace (classe em `client.superpython.core`), 9
add_editor() (método `client.superpython.core.Ace`), 9
annotate() (método `client.superpython.core.Ace`), 9

B

before_run() (método `client.superpython.core.Console`), 9

C

`client.superpython.core` (módulo), 9
Console (classe em `client.superpython.core`), 9

D

Dims (classe em `client.superpython.core`), 9
display_canvas() (método
 `ent.superpython.core.Console`), 9
display_saved() (método
 `ent.superpython.core.Console`), 9

G

get_content() (método `client.superpython.core.Ace`), 9

H

h (atributo `client.superpython.core.Dims`), 10

L

load() (método `client.superpython.core.SuperPython`), 10
logout_on_exit() (método
 `ent.superpython.core.SuperPython`), 10

M

main() (método `client.superpython.core.SuperPython`), 10

O

onexec_error() (método
 `ent.superpython.core.Console`), 9

S

save() (método `client.superpython.core.SuperPython`), 10

set_content() (método `client.superpython.core.Ace`), 9
setUp() (método `superpython_tests.SuperPythonTest`), 11
SuperPython (classe em `client.superpython.core`), 10
superpython_tests (módulo), 11
SuperPythonTest (classe em `superpython_tests`), 11

T

test_dirty() (método `client.superpython.core.Ace`), 9
test_main() (método
 `superpython_tests.SuperPythonTest`), 11

W

w (atributo `client.superpython.core.Dims`), 10
write() (método `client.superpython.core.Console`), 9

X

cli-x (atributo `client.superpython.core.Dims`), 10

Y

y (atributo `client.superpython.core.Dims`), 10