
SuperPython Documentation

Versão 1.1.3

Carlo Oliveira

03 set, 2018

1	SuperPython - Introdução	3
2	SuperPython - Manual	5
3	SuperPython - Módulos	7
4	Notas de Lançamento V. 1.1.3	9
4.1	Milestone	9
4.2	Aspectos do Lançamento	9
4.3	Melhoramentos	9
4.4	Consertos	10
4.5	Questões e Problemas Conhecidos	10
4.6	Lançamentos Anteriores e Posteriores	10
5	Lançamentos Anteriores	11
5.1	Notas de Lançamento V. 1.1.2	11
5.2	Notas de Lançamento V. 1.1.1	12
5.3	Notas de Lançamento V. 0.1.0	13
6	SuperPython - Módulos Principais	15
6.1	Módulo Cliente	15
6.2	Módulo Servidor	17
7	SuperPython - Módulos de Teste	19
8	Indices and tables	21
	Índice de Módulos do Python	23

Contents:

CAPÍTULO 1

SuperPython - Introdução

Este ambiente permite rodar um game stand alone e criar módulos que não estão no menu.

Para rodar um game basta chamar `<projeto>.is-by.us/code/_<modulo>`. O submódulo `main.py` será importado.

Para criar ou acessar um módulo fora do menu basta clicar a letra **O** no canto inferior direito do menu principal.

A biblioteca Phaser está disponível baixada do CDN. Use: `from browser import window; window.Phaser`

O projeto que se está usando pode ser caracterizado como a primeira palavra da url.

Pode se importar novos módulos adicionando o prefixo (módulo) `_spy` na frente do nome do módulo que se quer importar.

SuperPython - Modulos

SuperPython é programado em [Brython](#)

Funcionalidades Documentadas:

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

SuperPython

4.1 Milestone

hidenseek - Suporte a importação cruzada

4.2 Aspectos do Lançamento

4.2.1 Destaques dos Aspectos

Este ambiente permite criar novos módulos ou arquivos.

4.2.2 Aspecto #1

Para criar ou acessar um módulo fora do menu basta clicar a letra **O** no canto inferior direito do menu principal.

4.3 Melhoramentos

Novo menu visual e gif de carregamento.

4.3.1 Melhoramento #1

O novo menu visual é simétrico e suporta até cinquenta usuários.

4.3.2 Melhoria #2

Um gif animado com engrenagens rodando indica a carga do editor.

4.4 Consertos

O login, a url de projeto, novos módulos e importação de módulos foram consertados

4.4.1 Conserto #01

Quando se loga em um módulo ele fica tarjado no menu principal.

4.4.2 Conserto #01

Quando se loga em um módulo ele fica tarjado no menu principal.

4.4.3 Conserto #02

O projeto que se está usando pode ser caracterizado como a primeira palavra da url.

4.4.4 Conserto #03

Pode se criar e editar novos módulos e ou arquivos pelo menu extra principal.

4.4.5 Conserto #04

Pode se importar novos módulos adicionando o prefixo (módulo) `_spy` na frente.

4.5 Questões e Problemas Conhecidos

1. Ainda está congelado na versão antiga do Brython 3.0.2
2. Aparece uma imagem espúria *em construção* quando roda o jogo *stand alone*.

4.6 Lançamentos Anteriores e Posteriores

Próximo Lançamento: A ser definido *Lançamento 1.1.4*

Lançamento Anterior: Novembro 2015 *Lançamento 1.1.2*

5.1 Notas de Lançamento V. 1.1.2

SuperPython

5.1.1 Milestone

hidenseek - Suporte a importação cruzada

5.1.2 Aspectos do Lançamento

Destaques dos Aspectos

Este ambiente permite rodar um game stand alone e criar módulos que não estão no menu. Agregando a lib do Phaser.

Aspecto #1

Para rodar um game basta chamar <projeto>.is-by.us/code/_<modulo>. O submódulo *main.py* será importado.

Aspecto #2

Para criar ou acessar um módulo fora do menu basta clicar a letra **O** no canto inferior direito do menu principal.

Aspecto #3

A biblioteca Phaser está disponível baixada do CDN.

5.1.3 Melhoramentos

Novo menu visual e gif de carregamento.

Melhoramento #1

O novo menu visual é simétrico e suporta até cinquenta usuários.

Melhoramento #2

Um gif animado com engrenagens rodando indica a carga do editor.

5.1.4 Consertos

Ajuste da estrutura de configuração para facilitar os testes

Conserto #01

Adiciona o arquivo *vendor.py* para eliminar lib no import do bottle.

5.1.5 Questões e Problemas Conhecidos

1. Ainda está congelado na versão antiga do Brython 3.0.2
2. Aparece uma imagem espúria *em construção* quando roda o jogo *stand alone*.

5.1.6 Lançamentos Anteriores e Posteriores

Próximo Lançamento: A ser definido *Lançamento 1.1.3*

Lançamento Anterior: Novembro 2015 *Lançamento 1.1.1*

5.2 Notas de Lançamento V. 1.1.1

SuperPython

5.2.1 Milestone

hidenseek - Suporte a importação cruzada

5.2.2 Aspectos do Lançamento

Destaques dos Aspectos

Este ambiente permite a edição de códigos e importação do módulo de um outro projeto.

Aspecto #1

o nome `__name__` é atribuído «`__main__`» no programa principal.

5.2.3 Melhoramentos

Versão para uso do Google Application Engine e PyBuilder.

Melhoramento #1

Foi melhorada a 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.

5.2.4 Consertos

Conserto #01

o nome `__name__` é atribuído «`__main__`» no programa principal.

Conserto #02

A carga do módulo principal é feita por AJAX, evitando o congelamento do HTML.

5.2.5 Questões e Problemas Conhecidos

A funcionalidade ainda é muito simples, requer melhorias.

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

5.2.6 Lançamentos Anteriores e Posteriores

Próximo Lançamento: Maio 2016 *Lançamento 1.1.2*

Lançamento Anterior: Julho 2015 *Lançamento 0.1.0*

5.3 Notas de Lançamento V. 0.1.0

SuperPython

5.3.1 Milestone

Bruce - Protótipo da interface Gráfica

5.3.2 Aspectos do Lançamento

Destaques dos Aspectos

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

Aspecto #1

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

5.3.3 Melhoramentos

Versão para uso do Google Application Engine e PyBuilder.

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.

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.

5.3.4 Consertos

Nenhum conserto notável.

5.3.5 Questões e Problemas Conhecidos

A funcionalidade ainda é muito simples, requer melhorias.

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

5.3.6 Lançamentos Anteriores e Posteriores

Próximo Lançamento: A ser definido *Lançamento 0.2.0*

6.1 Módulo Cliente

6.1.1 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** – Brythom 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

before_run ()

display_canvas (*display='block'*)

display_saved (*message='SAVED'*)

onexec_error ()

write (*data*)

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

Bases: tuple

h

Alias for field number 3

w

Alias for field number 2

x

Alias for field number 0

y

Alias for field number 1

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

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*)

`client.superpython.core.main` (*browse, canvas, edit, projeto*)

Cria uma instância da classe Super Python.

Parâmetros

- **browse** – Módulo browser do Brython.
- **canvas** – Div *pydiv* onde se desenha gráficos.
- **edit** –
- **projeto** – Nome do projeto0.

Retorno instância da classe Super Python.

Veja também:

Module `client.superpython.core`

Nota: Unidade de Modelo Cliente.

6.2 Módulo Servidor

6.2.1 Controlador Principal

Veja também:

Module `server.controllers.main_controller`

Nota: Controlador principal da funcionalidade Web.

6.2.2 Controlador de Código

Veja também:

Module `server.controllers.main_controller`

Nota: Controla também a invocação de game stand alone.

6.2.3 Controlador de Edição

Veja também:

Module `server.controllers.main_controller`

Nota: Controla também a invocação de módulos `__init__`.

SuperPython - Módulos de Teste

Veja também:

Modules *SuperPython - Módulos Principais*

Nota: Módulo principal de testes unitários.

Veja também:

Module `spyweb_tests`

Nota: Unidade de Teste da funcionalidade Web.

CAPÍTULO 8

Indices and tables

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

C

`client.superpython.core` (*Web*), 15

A

Ace (classe em `client.superpython.core`), 15
add_editor() (método `client.superpython.core.Ace`), 15
annotate() (método `client.superpython.core.Ace`), 15

B

before_run() (método `client.superpython.core.Console`), 16

C

`client.superpython.core` (módulo), 15
Console (classe em `client.superpython.core`), 15

D

Dims (classe em `client.superpython.core`), 16
display_canvas() (método `ent.superpython.core.Console`), 16
display_saved() (método `ent.superpython.core.Console`), 16

G

get_content() (método `client.superpython.core.Ace`), 15

H

h (atributo `client.superpython.core.Dims`), 16

L

load() (método `client.superpython.core.SuperPython`), 16
logout_on_exit() (método `client.superpython.core.SuperPython`), 16

M

main() (método `client.superpython.core.SuperPython`), 16
main() (no módulo `client.superpython.core`), 16

O

onexec_error() (método `client.superpython.core.Console`), 16

S

save() (método `client.superpython.core.SuperPython`), 16
set_content() (método `client.superpython.core.Ace`), 15
SuperPython (classe em `client.superpython.core`), 16

T

test_dirty() (método `client.superpython.core.Ace`), 15

W

w (atributo `client.superpython.core.Dims`), 16
write() (método `client.superpython.core.Console`), 16

X

x (atributo `client.superpython.core.Dims`), 16

Y

cli-y (atributo `client.superpython.core.Dims`), 16