

Desenvolvendo com Python e EFL



Python EFL
Desenvolvimento com Python e EFL.



- Python
 - O que é ?
 - Por que ?
 - Exemplos de código
 - Dá pra escrever programas gráficos com Python?
 - Exemplos de programas gráficos
 - Python é uma linguagem séria ou de brinquedo?

- EFL (Enlightenment Foundation Libraries)
 - O que é ?
 - Por que ?
 - Maturidade
 - Exemplos
 - Visão geral da arquitetura
 - Ecore
 - Evas
 - Edje
 - Etk



- Python é uma linguagem de programação de alto nível, interpretada, interativa, orientada a objetos, de tipagem dinâmica e forte. Foi lançada por Guido van Rossum em 1991. Atualmente possui um modelo de desenvolvimento comunitário, aberto e gerenciado por uma organização sem fins lucrativos, a Python Software Foundation.
- A linguagem foi projetada com a filosofia de enfatizar a importância do esforço do programador sobre o esforço computacional. Prioriza a legibilidade do código sobre a velocidade ou expressividade. Combina uma sintaxe concisa e clara com os recursos poderosos de sua biblioteca padrão e por módulos e frameworks desenvolvidos por terceiros.

Fonte: Wikipédia

- Altíssimo nível: A forma de trabalhar é aproximada do raciocínio humano, por isso a curva de aprendizado é menor.
- Produtividade: por ter uma boa legibilidade de código, acaba-se tendo menos problemas em um processo de desenvolvimento de integração contínua.



- Linguagem glue: Linguagem que permite uma fácil integração com outras linguagens e ou componentes de software independente da linguagem que o componente seja escrito.
- Exemplo: Se você encontrou um projeto escrito em C/C++ e feito para ser utilizado com C/C++, python possui ferramentas para facilitar a criação de “**binding**” para utilizar os componentes do projeto C/C++ dentro da linguagem python.
- Entre outros motivos... :-)

Python: Exemplos de código

Alo mundo..

```
print('Ola, mundo!')
```

Imprimindo uma lista ordenada com os itens com a primeira letra em maiúscula...

```
lista = ['laranja', 'banana', 'uva']  
lista.sort()
```

```
for item in lista:  
    print item.capitalize()
```

Contando palavras em um arquivo

```
arquivo = file('text.txt')  
palavras = arquivo.read().split()  
unicas = set(palavras)
```

```
print 'Palavras: %d. Tirando as repetidas: %d' % (len(palavras), len(unicas))
```

Enviando email..

```
import smtplib
server = smtplib.SMTP('mail.xx.net')

# Envio de e-mail da conta @xy para a conta @xx

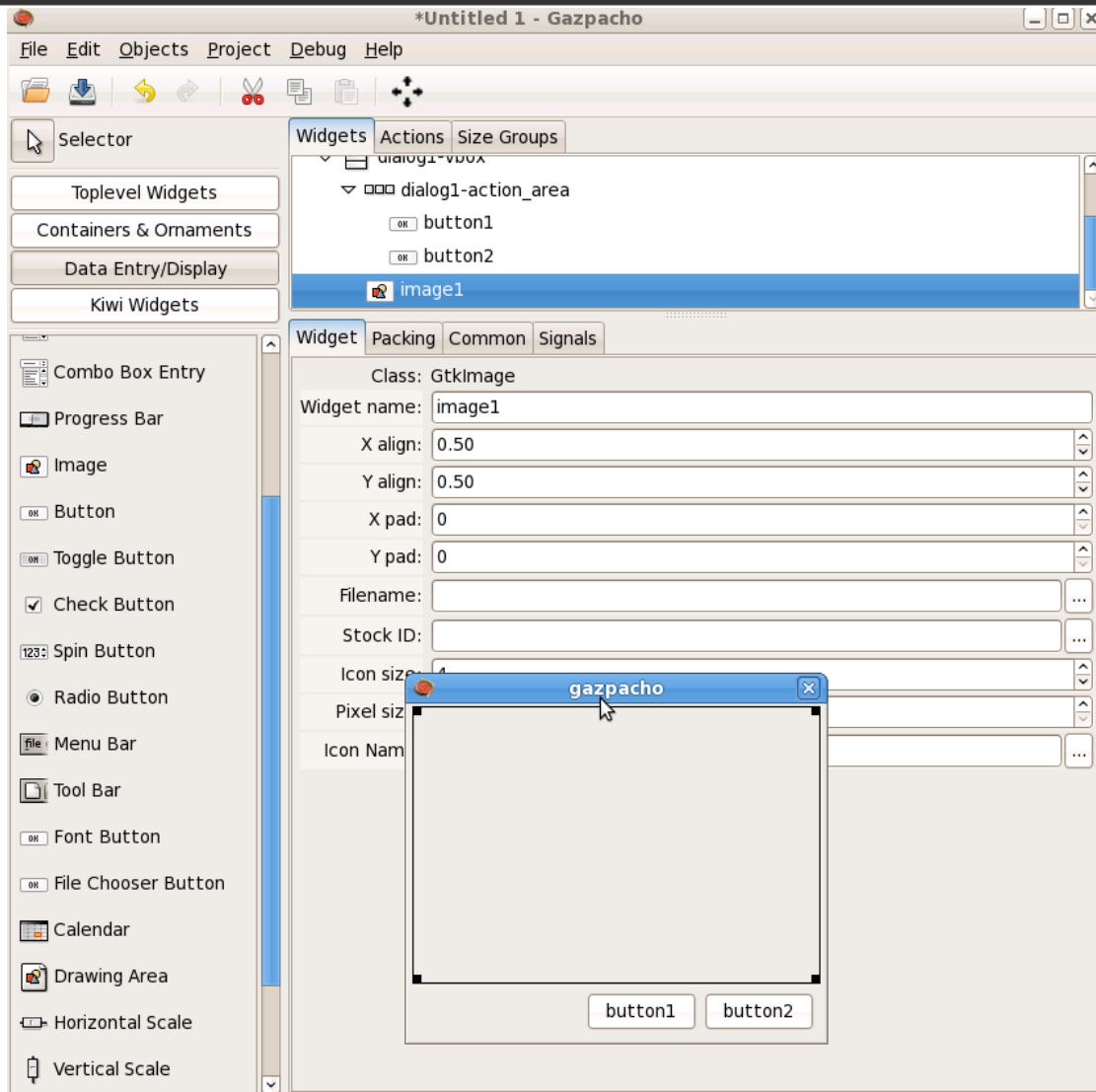
server.sendmail('leonardo@xy.com', 'leonardo@xx.net',
"""To: leonardo@xx.net
From: leonardo@xy.com
Subject: Teste de envio de email
Estou lhe enviando este email como um teste.
""")

server.quit()
print 'Envio OK'
```

Dá pra escrever programas gráficos com Python?

- Sim, e as opções são tantas que podem até atrapalhar. Pode-se utilizar quase a totalidade dos widgetsets disponíveis para Linux em Python e em Windows é possível utilizar a MFC e as APIs de mais alto nível desta plataforma.
- Pode-se utilizar: **EFL**, wxPython, PyGTK, PyQt, PythonCard, Tkinter, PyGlade, PMW, Kiwi, ncurses, OpenGL. Tem também: PIL, Vpython.

Exemplos de programas gráficos



Python é uma linguagem séria ou de brinquedo?

- Python não é uma linguagem de brinquedo se isso significar que é impossível fazer aplicações de uso real com ela, como editores ou desktops. Aliás, muito pelo contrário. Diversas empresas utilizam Python de maneira 'séria' para ligar seus muitos componentes. Python é utilizada como linguagem de scripting em diversos programas sérios, como jogos, renderizadores (desses que fazem filmes de sucesso). Se Python não fosse séria - nesse sentido - a NASA e a OTAN não a usariam.

Python é uma linguagem séria ou de brinquedo?

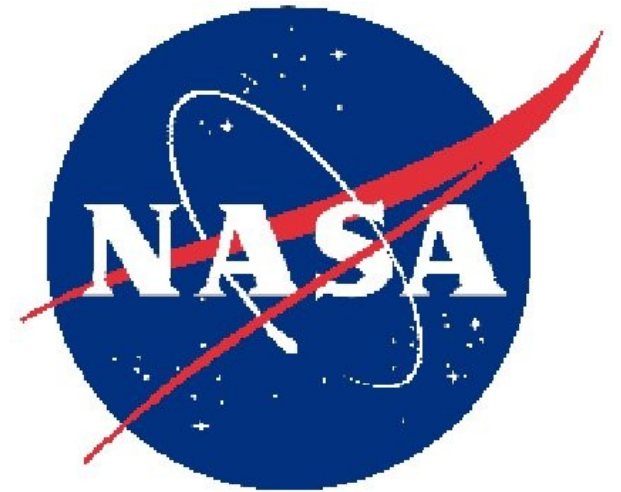
- Python é uma linguagem séria, ótima para quem está iniciando e, melhor ainda, poderosa para quem já é programador. Algo difícil de se obter em uma linguagem, diga-se de passagem. É possível se desenvolver qualquer tipo de aplicação em Python. É uma linguagem que dá liberdade ao programador e faz com que ele tenha que se preocupar em resolver os seus problemas e não se preocupar em deixar o compilador "feliz".

Python é uma linguagem séria ou de brinquedo?

- O Blender é um excelente exemplo de programa gráfico que utiliza python no desenvolvimento de seu código. Permite inclusive a facilidade de se inserir plug-ins através desta linguagem.

Exemplos de empresas sérias que usam python

Google™



YAHOO!®

NOKIA
Connecting People



EFL

EFL (Enlightenment Foundation Libraries)

- Originou-se no Enlightenment, que inicialmente(1997) tinha interesse em ser um gerenciador de janelas. Foi até incorporado ao gnome (1998) quando fundado pela Red Hat.
- O EFL nasceu junto com o Enlightenment 17 (E17), quando o objetivo do desenvolvimento do enlightenment deixou de ser apenas um gerenciador de janelas e passou a ser um ambiente desktop completo.

- Enlightenment 0.12 e 0.13



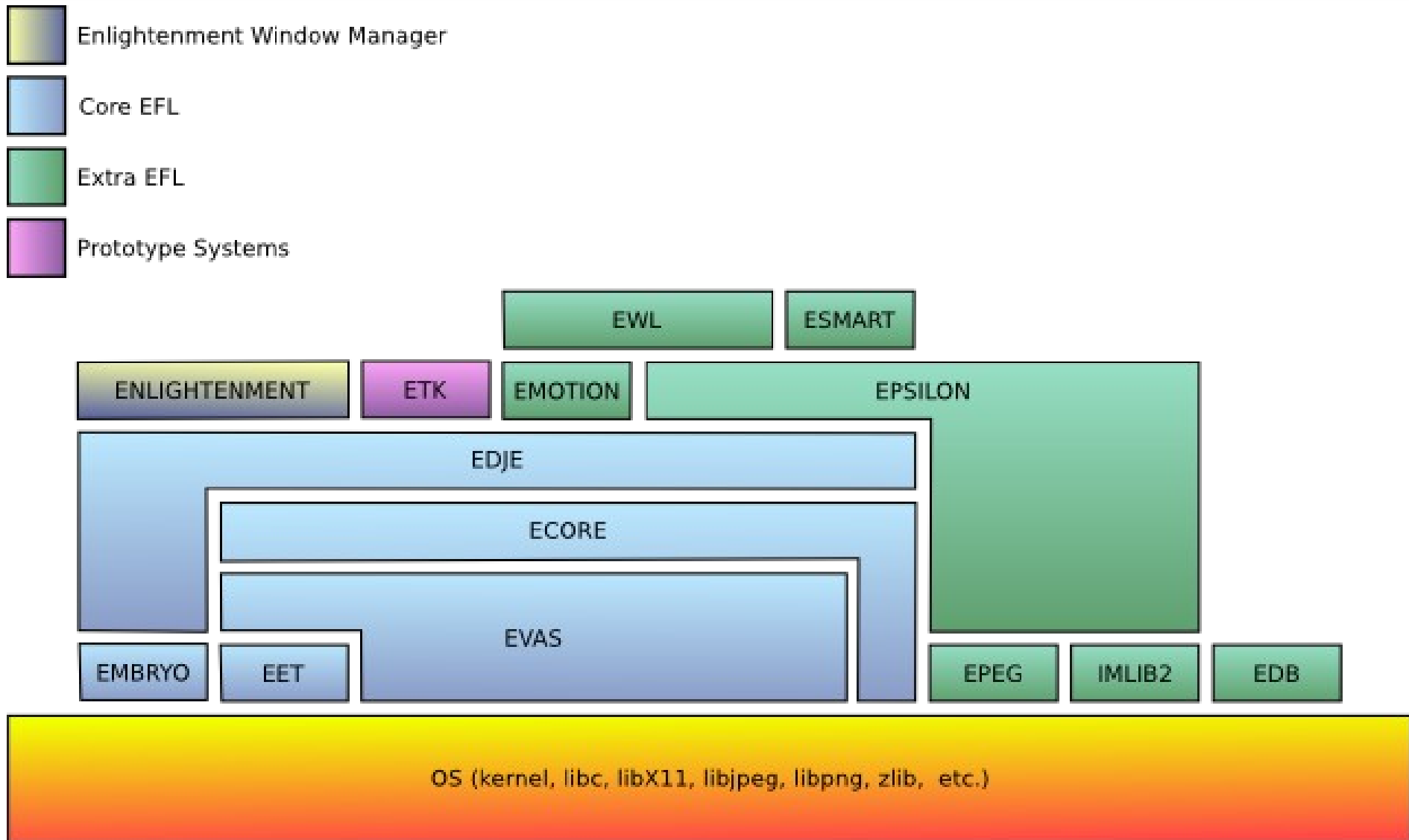
- Windows 95



EFL (Enlightenment Foundation Libraries)

- Foco:
 - Animação
 - Velocidade
- Componentes:
 - Eina
 - Eet
 - Evas
 - Ecore
 - Embryo
- Componentes:
 - Edje
 - Edbus
 - Efreet
 - Emotion
 - Epsilon

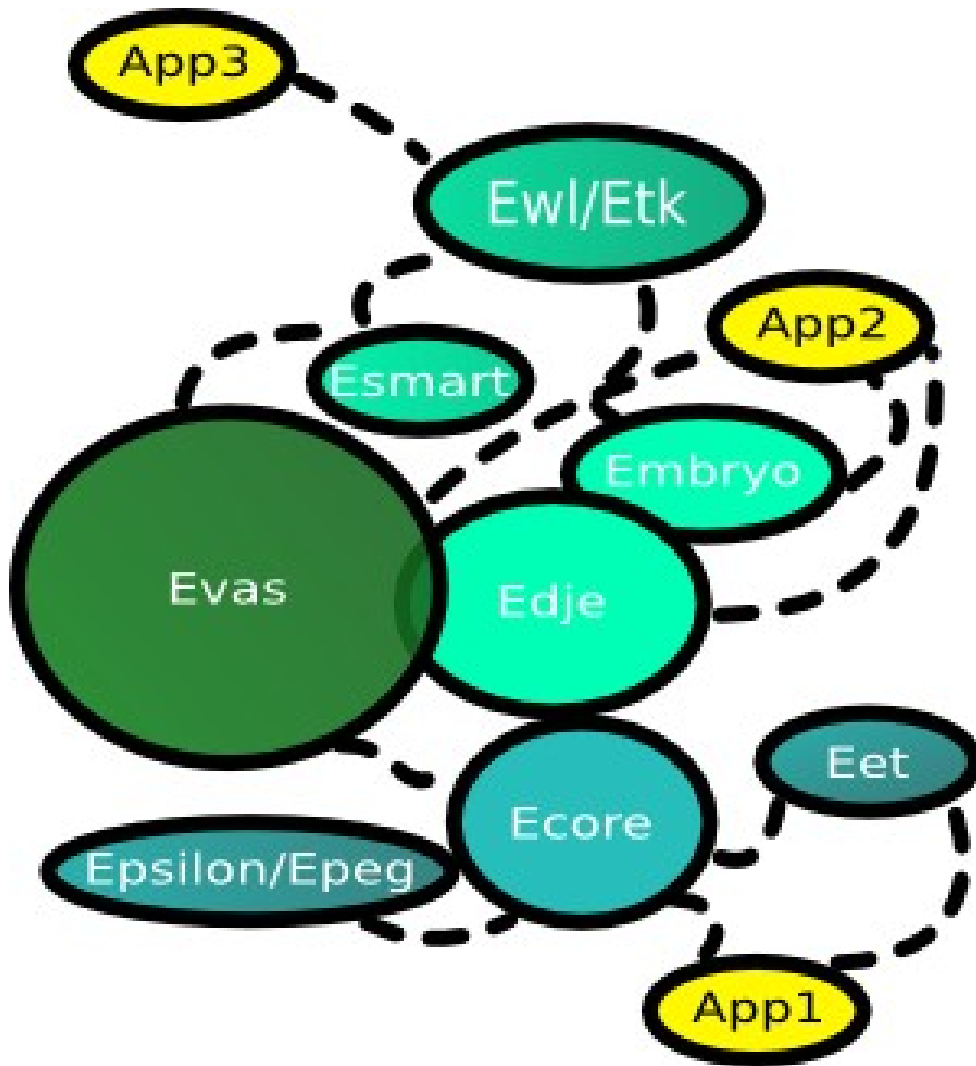
EFL: Visão geral



EFL: Breve descrição dos componentes

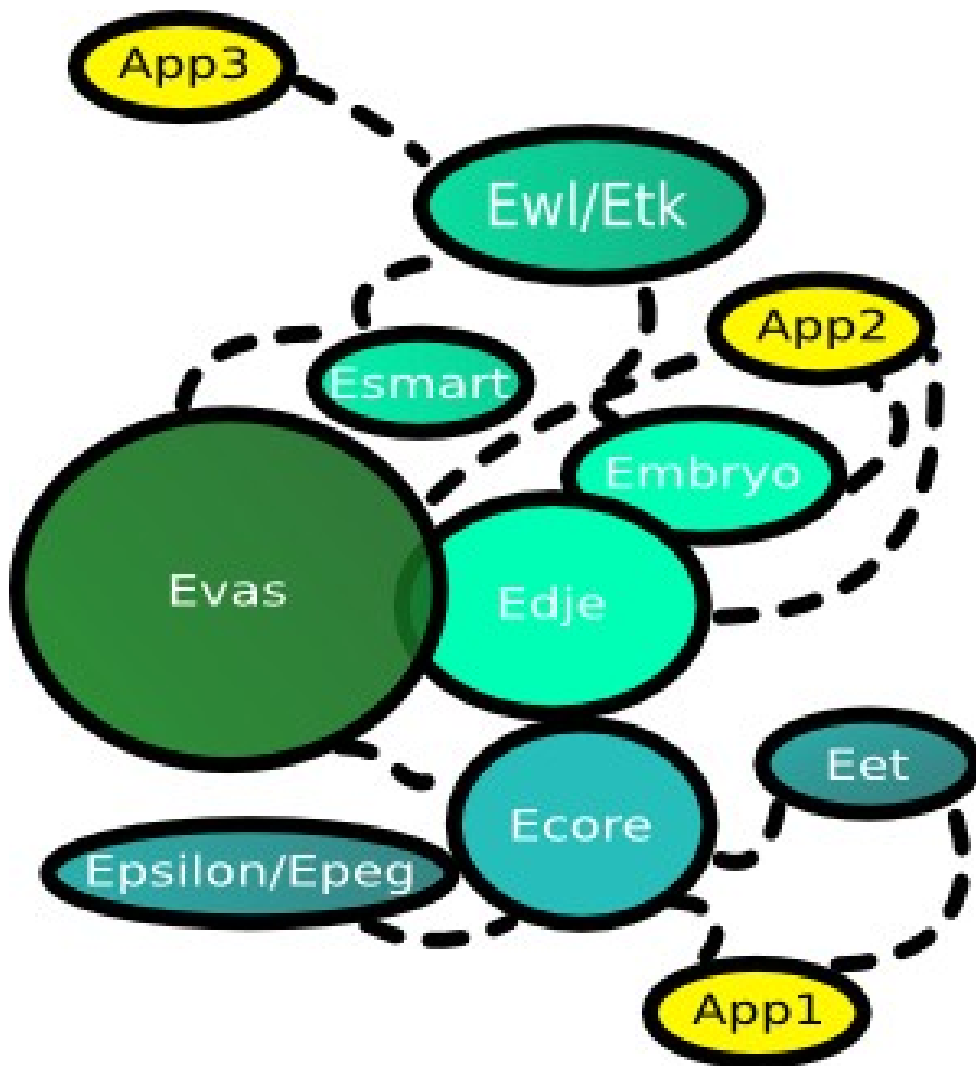
- **Eina:** Estruturas de dados/tipos.
- **Eet:** Armazenamento tipo arquivo, com compressão por chave.
- **Evas:** Canvas orientado a objetos que mantem estados.
- **Ecore:** Mainloop e acesso X11.
- **Embryo:** pequena linguagem de scripting.
- **Edje:** Poderoso engine de temas.
- **E_dbus:** integração com o Dbus.
- **Emotion:** biblioteca de player (xine,gstreamer)
- **Epsilon:** Gerador de thumbnails.

EFL: Visão do programador



- Note que as apps estão diferentemente acopladas a EFL.
- Acontece que as EFL são independentes entre si. Portanto você pode ter aplicações console usando as bibliotecas do EFL.

EFL: Visão do programador



- Fica claro na imagem ao lado que o Evas é o coração das EFL.

- Evas é baseado na imlib. O imlib 1 e 2 foram projetados e primariamente desenvolvidos por Rasterman(Carsten Haitzler).
- Eva nasceu no topo da imlib2 e possui as seguintes características:
 - Renderização retida. Canvas orientado a objetos que contem e mantem estados.
 - Baseado em abstração de engine: opengl, xrender e etc.
 - Velocidade. Desenha toda a cena e é rápido para pintar os pixels na tela.

- Abstrai a necessidade de conhecer a fundo manipulação de gráficos, não é preciso escovar bits.
- Projetada também para fácil incorporação de widgets.
- O tipo de renderização feita pelo Evas é o tipo **retido**. (Menor controle e flexibilidade). Porém as otimizações podem ser feitas pelo próprio canvas(Evas).

- Orientado a imagens: O que isso significa ?
 - Ao invés de guardar matrizes de pixels para serem pintados(renderizados) na tela, o que atualmente um canvas baseado em pixels faz. Guarda-se uma representação matemática da imagem.
 - Assim toda vez que a imagem é carregada para ser visualizada, o computador computa a representação matemática e mostra. Assim tornando-se independente de resolução e canvas.
 - Um exemplo de formato que utiliza vetores gráficos para armazenar é o SVG. É notável que independentemente do tamanho ele nunca perde a “qualidade”. Por que ?!

- Acesso a primitivas do X11, amigável com posix e unix.
- Controle de mainloop, api genérica para registro de *file descriptors*.
- Mainloop baseado em eventos, programação reativa.
- Sistema dorme enquanto não fd handlers ou timers.
- Controle de timers e animators.

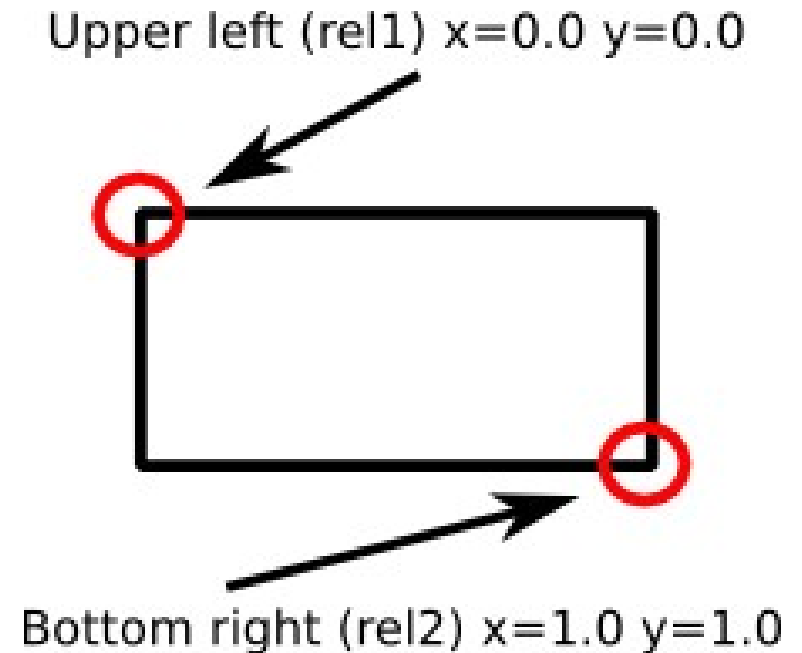
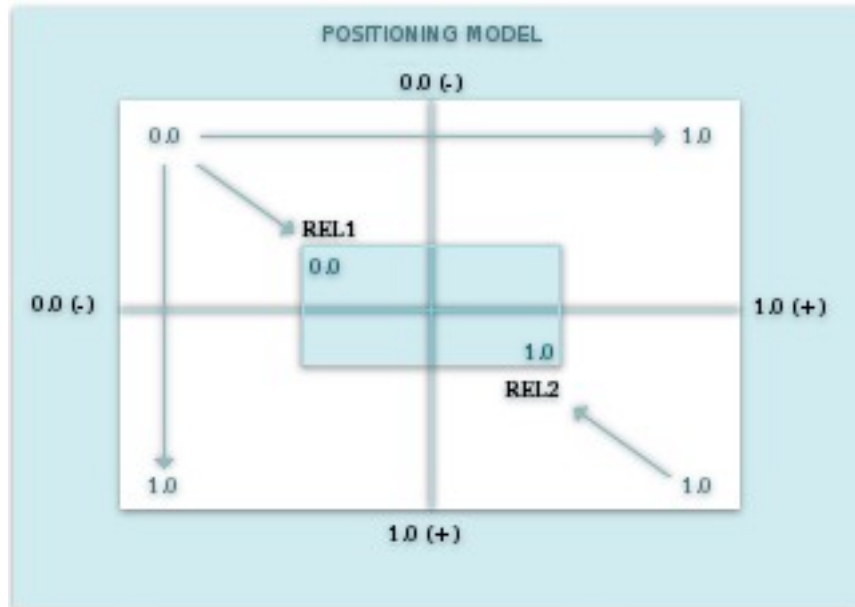
- Ecore-x: integração X11, despacho de eventos.
- Ecore-sdl: integração com SDL.
- Ecore-con: sockets, curl, dns, multicast...
- Ecore-wince: windows integration
- Ecore-ipc: Comunicação entre processos
- E outros.

- Abstrai configuração do canvas(Evas)
- Integra o mainloop
- Facilita o uso de eventos de entrada e eventos de janela.
- Possui mecanismo que automatizam algumas tarefas.(Exemplo pedir para Evas renderizar quando o sistema estiver em espera)

- É uma poderosa engine de temas, graças ao edje é possível separar lógica de aplicação da lógica de interface.
- Dependendo da finalidade da aplicação o edje pode ser considerado:
 - Engine de temas
 - Uma biblioteca de efeitos e animações gráficas.
 - Uma linguagem de descrição de interface
 - Uma abstração do Evas.

- A primeira vista o Edje parece ser complicado e um trabalho desnecessário. Mas após um estudo mais aprofundado você percebe que é totalmente válido pois facilita o desenvolvimento e o entendimento para quem já desenvolve web.

- O que é preciso ter em mente:
 - Coordenadas





python™

+



EFL

- O EFL é nativamente escrito em C para ser utilizado em C. Porém, foram escritos bindings para sua utilização a partir de Python, os bindings para python ainda não contemplam todas as funcionalidades ou todas as libs do EFL. Porém contempla as mais importantes. As partes que realmente faz o “negócio” acontecer.
- Quem escreveu esses bindings foi o Gustavo Barbieri (Um dos gurus da comunidade de Python e EFL).

- O que é um canvas ?
 - Originalmente é uma área plana de pintura (aquela usada por pintores)
 - O termo foi adotado também na computação gráfica para o processo de desenha/pintar na tela do computador.
- Portanto o Evas é quem pinta os objetos na tela
- ? mas que objetos ?

Primitivas	Descrição rápida
Rect	Retângulo típico, útil para clipping também.
Image	A primitiva mais comum.
Text	Objeto de texto, uma linha apenas, não é multilinhas
TextBlock	Objeto de texto que suporta multilinhas.
Line	Objeto de linha. Com inicio e fim
Polygon	Uma forma de uso geral
Gradient	Linear, radial, retângular e etc

- Eventos são uma sinais disparados para aplicação por causa de alguma ação.
- Exemplo:
 - Click do mouse
 - Movimentação do mouse
 - Tecla digitada
 - E etc.
- Os eventos executam callbacks.
- Mas.. o que são callbacks ? (Como assim bial ?!?!)

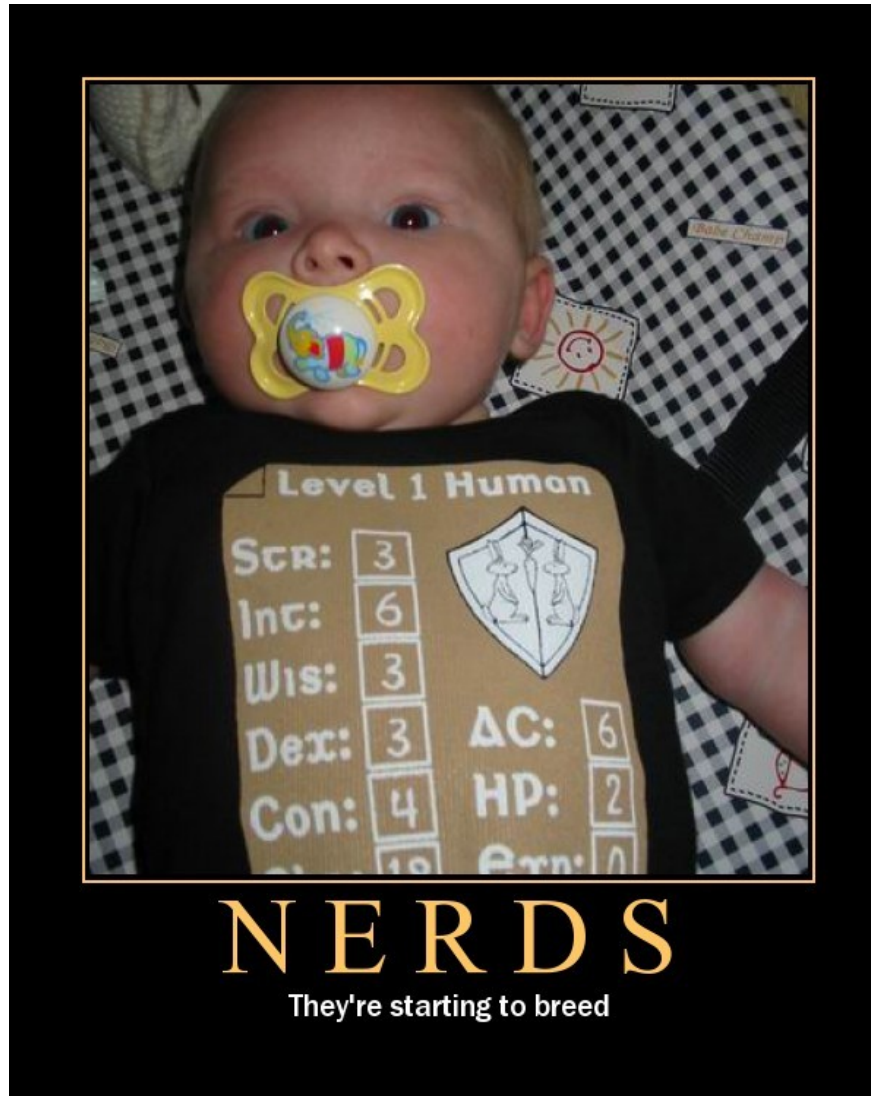
- São funções ou métodos executados a partir de um evento gerado.
- Para isso você precisa adicionar o callback desejado a um certo evento para executar o efeito a operação desejada.

- Exemplo evas
- Exemplo evas animado
- Outros exemplos
- Mostrar videos
- Softwares



- <http://homepages.pathfinder.gr/kazanaki/contrib/>
- <http://web.enlightenment.org/p.php?p=docs&l=en>
- <http://www.python.org.br>
- <http://wiki.enlightenment.org/>

Gostou ? Quer saber + ?



- Blog: <http://danielmartins.eti.br/blog>
- Email: root@danielmartins.eti.br