Cyber Mind

ALGORITMOS

Sumário

Java Script

Cyber Mind

JavaScript é uma linguagem de programação que permite implementar funcionalidades mais complexas em páginas web. Sempre que uma página web faz mais do que apenas mostrar informações estáticas para você - ela mostra em tempo real conteúdos atualizados, mapas interativos, animações gráficas em 2D/3D, vídeos, etc. - você pode apostar que o Javascript provavelmente está envolvido.

O caminho para o aprendizado

O JavaScript não é tão fácil de aprender como outras linguagens, como HTML e CSS, que são outros dois pilares do desenvolvimento front-end. Antes de tentar aprender JavaScript, é altamente recomendável que você aprenda e se familiarize com pelo menos estas duas tecnologias primeiro. Você pode começar através dos seguintes módulos:

Java Script

Cyber Mind

Começando na Web

Introdução ao HTML

Introdução ao CSS

Possuir experiência em outras linguagens de programação pode também ser útil.

Depois de aprender o básico de JavaScript, você estará apto a estudar tópicos mais avançados, como:

JavaScript aprofundado, como ensinado em Guia JavaScript

Referências API Web

Módulos

Este tópico contém os seguintes módulos, em uma ordem que sugerimos para estudá-los.

Java Script

Cyber Mind

Primeiros passos em JavaScript

Em nosso primeiro módulo JavaScript, primeiro responderemos algumas questões fundamentais como "o que é JavaScript?", "Como ele se parece?" E "o que ele pode fazer?", antes de passar para sua primeira experiência prática de escrever JavaScript. Depois disso, discutimos alguns recursos chave do JavaScript em detalhes, como variáveis, cadeias de caracteres, números e matrizes.

Blocos de codigo JavaScript

Neste módulo, continuaremos a falar sobre os principais recursos fundamentais do JavaScript, voltando nossa atenção para os tipos mais comuns de blocos de código, como instruções condicionais, funções e eventos. Você já viu essas coisas no curso, mas apenas de passagem, aqui discutiremos tudo explicitamente.

Java Script

Cyber Mind

Introdução a objetos em JavaScript

Em JavaScript, a maioria das coisas são objetos, desde seus principais recursos até as APIs do navegador. Você pode até criar seus próprios objetos.É importante entender a natureza orientada a objetos do JavaScript se você quiser ir mais longe com seu conhecimento da linguagem e escrever um código mais eficiente, portanto, fornecemos este módulo para ajudá-lo. Aqui ensinamos a teoria e a sintaxe de objetos em detalhes, observamos como criar seus próprios objetos e explicamos quais são os dados JSON e como trabalhar com eles.

JavaScript Assíncrono

Neste módulo, examinamos o JavaScript assíncrono, por que é importante e como ele pode ser usado para lidar efetivamente com possíveis operações de bloqueio, como a busca de recursos de um servidor.

Java Script

Cyber Mind

API's Web do lado cliente

Ao escrever JavaScript para sites ou aplicativos da Web, você não vai muito longe antes de começar a usar APIs - interfaces para manipular diferentes aspectos do navegador e do sistema operacional em que o site está sendo executado, ou até dados de outros sites ou serviços. Neste módulo, vamos explorar o que são as APIs e como usar algumas das APIs mais comuns que você encontrará com frequência em seu trabalho de desenvolvimento.

Aprofundando

Uma vez que você se acostumar com o mundo do JavaScript aqui estão alguns outros módulos em que você pode mergulhar:

Java Script

Cyber Mind

Biblioteca de referência JavaScript

Em nossa extensa biblioteca de referência você encontrará cada aspecto de Javascript tratado em detalhes: objetos globais, operadores, declarações e funções.

- Introdução à Orientação a Objetos (OO) em JavaScript

- Introdução aos conceitos de Programação orientada a objetos em JavaScript.

- Resolvendo problemas comuns com Javascript

Use Javascript para resolver problemas comuns, este link proporciona contéudos explicativos de como usar o JavaScript para solucionar problemas muito comuns ao criar uma página da Web.

PYTHON

Cyber Mind

Python é uma linguagem de programação de alto nível, interpretada, de script, imperativa, orientada a objetos, funcional, de tipagem dinâmica e forte. Foi lançada por Guido van Rossum em 1991. Atualmente possui um modelo de desenvolvimento comunitário, aberto e gerenciado pela organização sem fins lucrativos Python Software Foundation. Apesar de várias partes da linguagem possuírem padrões e especificações formais, a linguagem como um todo não é formalmente especificada. O padrão de facto é a implementação CPython.

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.

PYTHON

Cyber Mind

Python é uma linguagem de propósito geral de alto nível, multiparadigma, suporta o paradigma orientado a objetos, imperativo, funcional e procedural. Possui tipagem dinâmica e uma de suas principais características é permitir a fácil leitura do código e exigir poucas linhas de código se comparado ao mesmo programa em outras linguagens. Devido às suas características, ela é principalmente utilizada para processamento de textos, dados científicos e criação de CGIs para páginas dinâmicas para a web. Foi considerada pelo público a 3ª linguagem "mais amada", de acordo com uma pesquisa conduzida pelo site Stack Overflow em 2018,[6] e está entre as 5 linguagens mais populares, de acordo com uma pesquisa conduzida pela RedMonk.

O nome Python teve a sua origem no grupo humorístico britânico Monty Python, criador do programa Monty Python's Flying Circus, embora muitas pessoas façam associação com o réptil do mesmo nome (em português, píton ou pitão).

PYTHON

Cyber Mind

História

Guido van Rossum, São Francisco, Califórnia

O Python foi concebido no final de 1989 por Guido van Rossum no Instituto de Pesquisa Nacional para Matemática e Ciência da Computação (CWI), nos Países Baixos, como um sucessor da ABC capaz de tratar exceções e prover interface com o sistema operacional Amoeba[9] através de scripts. Também da CWI, a linguagem ABC era mais produtiva que C, ainda que com o custo do desempenho em tempo de execução. Mas ela não possuía funcionalidades importantes para a interação com o sistema operacional, uma necessidade do grupo. Um dos focos primordiais de Python era aumentar a produtividade do programador.

Python foi feita com base na linguagem ABC, possui parte da sintaxe derivada do C, compreensão de listas, funções anonimas e função map de Haskell. Os iteradores são baseados na Icon, tratamentos de exceção e módulos da Modula-3, expressões regulares de Perl.

PYTHON

Cyber Mind

Em 1991, Guido publicou o código (nomeado versão 0.9.0) no grupo de discussão alt.sources. Nessa versão já estavam presentes classes com herança, tratamento de exceções, funções e os tipos de dado nativos list, dict, str, e assim por diante. Também estava presente nessa versão um sistema de módulos emprestado do Modula-3. O modelo de exceções também lembrava muito o do Modula-3, com a adição da opção else clause. Em 1994 foi formado o principal fórum de discussão do Python, comp.lang.python, um marco para o crescimento da base de usuários da linguagem.

A versão 1.0 foi lançada em janeiro de 1994. Novas funcionalidades incluíam ferramentas para programação funcional como lambda, map, filter e reduce. A última versão enquanto Guido estava na CWI foi o Python 1.2. Em 1995, ele continuou o trabalho no CNRI em Reston, Estados Unidos, de onde lançou diversas versões. Na versão 1.4 a linguagem ganhou parâmetros nomeados (a capacidade de passar parâmetro pelo nome e não pela posição na lista de parâmetros) e suporte nativo a números complexos, assim como uma forma de encapsulamento.

PYTHON

Cyber Mind

Ainda na CNRI, Guido lançou a iniciativa Computer Programming for Everybody (CP4E; literalmente, "Programação de Computadores para Todos"), que visava tornar a programação mais acessível, um projeto financiado pela DARPA. Atualmente o CP4E encontra-se inativo.

Em 2000, o time de desenvolvimento da linguagem se mudou para a BeOpen a fim de formar o time PythonLabs. A CNRI pediu que a versão 1.6 fosse lançada para marcar o fim de desenvolvimento da linguagem naquele local. O único lançamento na BeOpen foi o Python 2.0, e após o lançamento o grupo de desenvolvedores da PythonLabs agrupou-se na Digital Creations.

Python 2.0 implementou list comprehension, uma relevante funcionalidade de linguagens funcionais como SETL e Haskell. A sintaxe da linguagem para essa construção é bastante similar a de Haskell, exceto pela preferência do Haskell por caracteres de pontuação e da preferência do python por palavras reservadas alfabéticas. Essa versão 2.0 também introduziu um sistema coletor de lixo capaz de identificar e tratar ciclos de referências.

PYTHON

Cyber Mind

Já o 1.6 incluiu uma licença CNRI substancialmente mais longa que a licença CWI que estavam usando nas versões anteriores. Entre outras mudanças, essa licença incluía uma cláusula atestando que a licença era governada pelas leis da Virgínia. A Free Software Foundation alegou que isso era incompatível com a GNU GPL. Tanto BeOpen quanto CNRI e FSF negociaram uma mudança na licença livre do Python que o tornaria compatível com a GPL. Python 1.6.1 é idêntico ao 1.6.0, exceto por pequenas correções de falhas e uma licença nova, compatível com a GPL.

Python 2.1 era parecido com as versões 1.6.1 e 2.0. Sua licença foi renomeada para Python Software Foundation License. Todo código, documentação e especificação desde o lançamento da versão alfa da 2.1 é propriedade da Python Software Foundation (PSF), uma organização sem fins lucrativos fundada em 2001, um modelo tal qual da Apache Software Foundation. O lançamento incluiu a mudança na especificação para suportar escopo aninhado, assim como outras linguagens com escopo estático. Esta funcionalidade estava desativada por padrão, e somente foi requerida na versão 2.2.

PYTHON

Cyber Mind

Uma grande inovação da versão 2.2 foi a unificação dos tipos Python (escritos em C) e classes (escritas em Python) em somente uma hierarquia. Isto tornou o modelo de objetos do Python consistentemente orientado a objeto. Também foi adicionado generator, inspirado em Icon.
O incremento da biblioteca padrão e as escolhas sintáticas foram fortemente influenciadas por Java em alguns casos: o pacote logging introduzido na versão 2.3,[18] o analisador sintático SAX, introduzido na versão 2.0 e a sintaxe de decoradores que usa @, adicionadas na versão 2.4.

Em 1 de outubro de 2008 foi lançada a versão 2.6, já visando a transição para a versão 3.0 da linguagem. Entre outras modificações, foram incluídas bibliotecas para multiprocessamento, JSON e E/S, além de uma nova forma de formatação de cadeias de caracteres.

PYTHON

Cyber Mind

Atualmente a linguagem é usada em diversas áreas, como servidores de aplicação e computação gráfica. Está disponível como linguagem de script em aplicações como OpenOffice (Python UNO Bridge), Blender e pode ser utilizada em procedimentos armazenados no sistema gerenciador de banco de dados PostgreSQL (PL/Python).

A terceira versão da linguagem foi lançada em dezembro de 2008, chamada Python 3.0 ou Python 3000. Com noticiado desde antes de seu lançamento, houve quebra de compatibilidade com a família 2.x para corrigir falhas que foram descobertas neste padrão, e para limpar os excessos das versões anteriores.[8] A primeira versão alfa foi lançada em 31 de agosto de 2007, a segunda em 7 de dezembro do mesmo ano.

Mudanças da versão incluem a alteração da palavra reservada print, que passa a ser uma função, tornando mais fácil a utilização de uma versão alternativa da rotina. Em Python 2.6, isso já está disponível ao adicionar o código from __future__ import print_function.[24] Também, a mudança para Unicode de todas as cadeias de caracteres.

PYTHON

Cyber Mind

Em 2012, foi criado o Raspberry Pi, cujo nome foi baseado na linguagem Python. Uma das principais linguagens escolhidas é Python. Python influenciou várias linguagens, algumas delas foram Boo e Cobra, que usa a indentação como definição de bloco e Go, que se baseia nos princípios de desenvolvimento rápido de Python.

Atualmente, Python é um dos componentes padrão de vários sistemas operacionais, entre eles estão a maioria das distribuições do Linux, AmigaOS 4, FreeBSD, NetBSD, OpenBSD e OS X. A linguagem se tornou a padrão no curso de ciências da computação do MIT em 2009.

Filosofia

Python 3. The standard type hierarchy.png

Parte da cultura da linguagem gira ao redor de The Zen of Python, um poema que faz parte do documento "PEP 20 (The Zen of Python)",[26] escrito pelo programador em Python de longa data Tim Peters, descrevendo sumariamente a filosofia do Python. Pode-se vê-lo através de um easter egg do Python pelo comando:

PYTHON

Cyber Mind

>>> import this

Construções

Construções de Python incluem: estrutura de seleção (if, else, elif); estrutura de repetição (for, while), que itera por um container, capturando cada elemento em uma variável local dada; construção de classes (class); construção de sub-rotinas (def); construção de escopo (with), como por exemplo para adquirir um recurso.

Tipos de dados

A tipagem de Python é forte, pois os valores e objetos têm tipos bem definidos e não sofrem coerções como em C ou Perl. São disponibilizados diversos tipos de dados nativos:

PYTHON

Cyber Mind

Tipo de dado	Descrição	Exemplo da sintaxe
str, unicode	Uma cadeia de caracteres imutável	'Wikipedia', u'Wikipedia'
list	Lista heterogênea mutável	[4.0, 'string', True]
tuple	Tupla imutável	(4.0, 'string', True)
set, frozenset	Conjunto não ordenado, não contém elementos duplicados	set([4.0, 'string', True])
Frozenset		([4.0, 'string', True])
dict	conjunto associativo	{'key1': 1.0, 'key2': False}
int	Número de precisão fixa, é transparentemente convertido para long caso não caiba em um	
int. 42		2147483648L
float	Ponto flutuante	3.1415927
complex Número complexo		3+2j
bool	Booleano	True ou False
!=	Diferente	Diferente

PYTHON

Cyber Mind

Python também permite a definição dos tipos de dados próprios, através de classes. Instâncias são construídas invocando a classe (FooClass()), e as classes são instância da classe type, o que permite metaprogramação e reflexão. Métodos são definidos como funções anexadas à classe, e a sintaxe instância.método(argumento) é um atalho para Classe.método(instância, argumento). Os métodos devem referenciar explicitamente a referência para o objeto incluindo o parâmetro self como o primeiro argumento do método.

Antes da versão 3.0, Python possuía dois tipos de classes: "old-style" e "new-style". Classes old-style foram eliminadas no Python 3.0, e todas são new-style. Em versões entre 2.2 e 3.0, ambos tipos de classes podiam ser usadas. A sintaxe de ambos estilos é a mesma, a diferença acaba sendo de onde objeto da classe é herdado, direta ou indiretamente (todas classes new-style herdam de object e são instancias de type). As classes new-styles nada mais são que tipos definidos pelo usuário.

PYTHON

Cyber Mind

Palavras reservadas

O Python 3 define as seguintes palavras reservadas:

False None True and as assert break class continue def del elif else except finally for from global if import in is lambda not nonlocal or pass raise try return while with yield

Operadores

Os operadores básicos de comparação como ==, <, >=, entre outros são usados em todos os tipos de dados, como números, cadeias de texto, listas e mapeamentos. Comparações em cadeia como a < b < c possuem o mesmo significado básico que na matemática: os termos são comparadas na ordem. É garantido que o processamento da expressão lógica irá terminar tão cedo o veredito seja claro, o princípio da avaliação mínima. Usando a expressão anterior, se a < b é falso, c não é avaliado.

PYTHON

Cyber Mind

Quanto aos operadores lógicos, até Python 2.2 não havia o tipo de dado booleano. Em todas as versões da linguagem os operadores lógicos tratam "", 0, None, 0.0, [] e {} como falso, enquanto o restante é tratado como verdadeiro de modo geral. Na versão 2.2.1 as constantes True e False foram adicionadas (subclasses de 1 e 0 respectivamente). A comparação binária retorna uma das duas constantes acima.

Os operadores booleanos and e or também seguem a avaliação mínima. Por exemplo, y == 0 or x/y > 100 nunca lançará a exceção de divisão por zero.

Interpretador interativo

O interpretador interativo é uma característica diferencial da linguagem, porque há a possibilidade de testar o código de um programa e receber o resultado em tempo real, antes de iniciar a compilação ou incluí-las nos programas. Por exemplo:

PYTHON

Cyber Mind

```
>>> 1+1
2
>>>
>>> a = 1+1
>>> print a
2
>>> print(a)
2
>>>
```

Nota: A partir da versão 3.0, o comando print passou a ser uma função, sendo obrigatório o uso de parênteses.

PYTHON

Cyber Mind

Análise léxica

Exemplo de script

No segundo capítulo do Manual de Referência da Linguagem Python é citado que a análise léxica é uma análise do interpretador em si, os programas são lidos por um analisador sintático que divide o código em tokens.

Todo programa é dividido em linhas lógicas que são separadas pelo token NEWLINE ou NOVA LINHA, as linhas físicas são trechos de código divididos pelo caractere ENTER. Linhas lógicas não podem ultrapassar linhas físicas com exceção de junção de linhas, por exemplo:

```
if resultado > 2 and \
 1 <= 5  and \
    2 < 5:
        print ('Resultado: %f' % d)
```

ou

PYTHON

Cyber Mind

```
MESES_DO_ANO = ['janeiro', 'fevereiro', 'março',
                'abril',   'maio',      'junho',
                'julho',   'agosto',    'setembro',
                'outubro', 'novembro',  'dezembro']
```

Para a delimitação de blocos de códigos, os delimitadores são colocados em uma pilha e diferenciados por sua indentação, iniciando a pilha com valor 0 (zero) e colocando valores maiores que os anteriores na pilha. Para cada começo de linha, o nível de indentação é comparado com o valor do topo da pilha. Se o número da linha for igual ao topo da pilha, a pilha não é alterada. Se o valor for maior, a pilha recebe o nível de indentação da linha e o nome INDENT (empilhamento). Se o nível de indentação for menor, então é desempilhado até chegar a um nível de indentação recebendo o nome DEDENT (desempilhamento). Se não encontrar nenhum valor, é gerado um erro de indentação.

PYTHON

Cyber Mind

Abaixo um exemplo de permutação, retirado do capítulo 2.1 sobre Estrutura de linhas na Análise léxica do Manual de Referência da linguagem (Language Reference Manual):

```
        def perm(l):                NOVA LINHA
INDENT      if len(l) <= 1:         NOVA LINHA
INDENT        return[1]             NOVA LINHA
DEDENT      r = [ ]                 NOVA LINHA
            for i in range(len(l)):  NOVA LINHA
INDENT        s = l[:i] + l[i+1:]   NOVA LINHA
              p = perm(s)           NOVA LINHA
DEDENT      for x in p:             NOVA LINHA
INDENT        r.append(l[i:i+1]+x)  NOVA LINHA
DEDENT      return r
```

PYTHON

Cyber Mind

Indentação

Python foi desenvolvido para ser uma linguagem de fácil leitura, com um visual agradável, frequentemente usando palavras e não pontuações como em outras linguagens. Para a separação de blocos de código, a linguagem usa espaços em branco e indentação ao invés de delimitadores visuais como chaves (C, Java) ou palavras (BASIC, Fortran, Pascal). Diferente de linguagens com delimitadores visuais de blocos, em Python a indentação é obrigatória. O aumento da indentação indica o início de um novo bloco, que termina da diminuição da indentação.

Usando um editor de texto comum é muito fácil existir erros de indentação, o recomendado é configurar o editor conforme a análise léxica do Python ou utilizar uma IDE. Todas as IDE que suportam a linguagem fazem indentação automaticamente.

PYTHON

Cyber Mind

Exemplo:

Indentação correta

```
def valor1():
    while True:
        try:
            c = int(input('Primeiro Valor: '))
            return c
        except ValueError:
            print 'Inválido!'
```

Indentação incorreta

```
def valor1():
while True:
try:
c = int(input('Primeiro Valor: '))
return c
except ValueError:
print 'Inválido!'
```

PYTHON

Cyber Mind

Indentação

Python foi desenvolvido para ser uma linguagem de fácil leitura, com um visual agradável, frequentemente usando palavras e não pontuações como em outras linguagens. Para a separação de blocos de código, a linguagem usa espaços em branco e indentação ao invés de delimitadores visuais como chaves (C, Java) ou palavras (BASIC, Fortran, Pascal). Diferente de linguagens com delimitadores visuais de blocos, em Python a indentação é obrigatória. O aumento da indentação indica o início de um novo bloco, que termina da diminuição da indentação.

Usando um editor de texto comum é muito fácil existir erros de indentação, o recomendado é configurar o editor conforme a análise léxica do Python ou utilizar uma IDE. Todas as IDE que suportam a linguagem fazem indentação automaticamente.

PYTHON

Cyber Mind

O código está correto para os dois exemplos, mas o analisador léxico verificará se a indentação está coerente. O analisador reconhecerá as palavras reservadas while, def, try, except, return, print e as cadeias de caracteres entre aspas simples e a indentação, e se não houver problemas o programa executará normalmente, senão apresentará a exceção: "Seu programa está com erro no bloco de indentação".

Na internet, há uma comparação de velocidade e de codificação entre as linguagens Python e BASIC, esta última, o dialeto BBC BASIC for Windows.

Compilador de bytecode

A linguagem é de altíssimo nível, como já dito, mas ela também pode compilar seus programas para que a próxima vez que o executar não precise compilar novamente o programa, reduzindo o tempo de carga na execução.

PYTHON

Cyber Mind

Utilizando o interpretador interativo não é necessário a criação do arquivo de Python compilado, os comandos são executados interativamente. Porém quando um programa ou um módulo é evocado, o interpretador realiza a análise léxica e sintática, compila o código de alto nível se necessário e o executa na máquina virtual da linguagem.

O bytecode é armazenado em arquivos com extensão .pyc ou .pyo, este último no caso de bytecode otimizado. Interessante notar que o bytecode da linguagem também é de alto nível, ou seja, é mais legível aos seres humanos que o código de byte do C, por exemplo. Para descompilar um código de byte é utilizado o módulo dis da biblioteca padrão da linguagem e existem módulos de terceiros que tornam o bytecode mais confuso, tornando a descompilação ineficaz.

PYTHON

Cyber Mind

Normalmente, o Python trabalha com dois grupos de arquivos:

Os módulos do núcleo da linguagem, sua biblioteca padrão e os módulos independentes, criados pelo usuário.

No núcleo do interpretador existe o analisador léxico, o analisador sintático que utiliza Estruturas de Objetos (tempo de execução), o Compilador que aloca memória (tempo de execução) e depois do Avaliador de código que modifica o estado atual do programa (tempo de execução), mostrando resultado para o usuário.

Orientação a objetos

Python suporta a maioria das técnicas da programação orientada a objeto. Qualquer objeto pode ser usado para qualquer tipo, e o código funcionará enquanto haja métodos e atributos adequados. O conceito de objeto na linguagem é bastante abrangente: classes, funções, números e módulos são todos considerados objetos. Também há suporte para metaclasses, polimorfismo, e herança (inclusive herança múltipla). Há um suporte limitado para variáveis privadas.

PYTHON

Cyber Mind

Na versão 2.2 de Python foi introduzido um novo estilo de classes em que objetos e tipos foram unificados, permitindo a especialização de tipos. Já a partir da versão 2.3 foi introduzido um novo método de resolução de ambiguidades para heranças múltiplas.

Uma classe é definida com class nome:, e o código seguinte é a composição dos atributos. Todos os métodos da classe recebem uma referência a uma instância da própria classe como seu primeiro argumento, e a convenção é que se chame este argumento self. Assim os métodos são chamados objeto.método(argumento1, argumento2, ...) e são definidos iguais a uma função, como método(self, argumento1, argumento2, ...). Veja que o parâmetro self conterá uma referência para a instância da classe definida em objeto quando for efetuada esta chamada. Os atributos da classe podem ser acessados em qualquer lugar da classe, e os atributos de instância (ou variável de instância) devem ser declarados dentro dos métodos utilizando a referência à instância atual (self) (ver código contextualizado em anexo).

PYTHON

Em Python não existe proteção dos membros duma classe ou instância pelo interpretador, o chamado encapsulamento. Convenciona-se que atributos com o nome começando com um _ são de uso privado da classe, mas não há um policiamento do interpretador contra acesso a estes atributos. Uma exceção são nomes começando com __, no caso em que o interpretador modifica o nome do atributo (ver código contextualizado em anexo).

Python permite polimorfismo, que condiz com a reutilização de código. É fato que funções semelhantes em várias partes do software sejam utilizadas várias vezes, então definimos esta função como uma biblioteca e todas as outras funções que precisarem desta a chamam sem a necessidade de reescrevê-la (ver código contextualizado em anexo).

Python não possui overloading; não é possível criar duas funções com o mesmo nome, pois elas são consideradas atributos da classe. Caso o nome da função se repita em outra assinatura, o interpretador considera esta última como override e sobrescreve a função anterior. Algumas operações entre diferentes tipos são realizadas através de coerção (ex.: 3.2 + 3).

PYTHON

É possível encapsular abstrações em módulos e pacotes. Quando um arquivo é criado com a extensão .py, ele automaticamente define um módulo. Um diretório com vários módulos é chamado de pacote e deve conter um modulo chamado __init__, para defini-lo como principal. Estas diferenciações ocorrem apenas no sistema de arquivos. Os objetos criados são sempre módulos. Caso o código não defina qual dos módulos será importado, o padrão é o __init__.

Programação funcional

Uma das construções funcionais de Python é compreensão de listas, uma forma de construir listas. Por exemplo, pode-se usar a técnica para calcular as cinco primeiras potências de dois. O algoritmo quicksort também pode ser expressado usando a mesma técnica (ver códigos contextualizados para ambos os casos em anexo).

Em Python, funções são objetos de primeira classe que podem ser criados e armazenados dinamicamente. O suporte a funções anônimas está na construção lambda (cálculo Lambda). Não há disponibilidade de funções anônimas de fato, pois os lambdas contêm somente expressões e não blocos de código.

PYTHON

Cyber Mind

Python também suporta clausuras léxicas desde a versão 2.2 (ver códigos contextualizados para ambos os casos em anexo). Já geradores foram introduzidos na versão 2.2 e finalizados na versão 2.3, e representam o mecanismo de Python para a avaliação preguiçosa de funções (ver códigos contextualizados para ambos os casos em anexo).

Tratamento de exceções

Python suporta e faz uso constante de tratamento de exceções como uma forma de testar condições de erro e outros eventos inesperados no programa. É inclusive possível capturar uma exceção causada por um erro de sintaxe. O estilo da linguagem apóia o uso de exceções sempre que uma condição de erro pode aparecer. Por exemplo, ao invés de testar a disponibilidade de acesso a um recurso, a convenção é simplesmente tentar usar o recurso e capturar a exceção caso o acesso seja rejeitado (recurso inexistente, permissão de acesso insuficiente, recurso já em uso, ...).

PYTHON

Cyber Mind

Exceções são usadas frequentemente como uma estrutura de seleção, substituindo blocos if-else, especialmente em situações que envolvem threads. Uma convenção de codificação é o EAFP, do inglês, "é mais fácil pedir perdão que permissão". Isso significa que em termos de desempenho é preferível capturar exceções do que testar atributos antes de os usar. Segue abaixo exemplos de código que testam atributos ("pedem permissão") e que capturam exceções ("pedem perdão"):

Teste de atributo

```
if hasattr(spam, 'eggs'):
    ham = spam.eggs
else:
    handle_error()
```

Captura de exceção

```
try:
    ham = spam.eggs
except AttributeError:
    handle_error()
```

PYTHON

Cyber Mind

Ambos os códigos produzem o mesmo efeito, mas há diferenças de desempenho. Quando spam possui o atributo eggs, o código que captura exceções é mais rápido. Caso contrário, a captura da exceção representa uma perda considerável de desempenho, e o código que testa o atributo é mais rápido. Na maioria dos casos o paradigma da captura de exceções é mais rápido, e também pode evitar problemas de concorrência. Por exemplo, num ambiente multitarefa, o espaço de tempo entre o teste do atributo e seu uso de fato pode invalidar o atributo, problema que não acontece no caso da captura de exceções.

Biblioteca padrão

Python possui uma grande biblioteca padrão, geralmente citada como um dos maiores trunfos da linguagem, fornecendo ferramentas para diversas tarefas. Por conta da grande variedade de ferramentas fornecida pela biblioteca padrão, combinada com a habilidade de usar linguagens de nível mais baixo como C e C++, Python pode ser poderosa para conectar componentes diversos de software.

PYTHON

Cyber Mind

A biblioteca padrão conta com facilidades para escrever aplicações para a Internet, contando com diversos formatos e protocolos como MIME e HTTP. Também há módulos para criar interfaces gráficas, conectar em bancos de dados relacionais e manipular expressões regulares.

Algumas partes da biblioteca são cobertas por especificações (por exemplo, a implementação WSGI da wsgiref segue o PEP 333[33]), mas a maioria dos módulos não segue.

Interoperabilidade

Um outro ponto forte da linguagem é sua capacidade de interoperar com várias outras linguagens, principalmente código nativo. A documentação da linguagem inclui exemplos de como usar a Python C-API para escrever funções em C que podem ser chamadas diretamente de código Python - mas atualmente esse sequer é o modo mais indicado de interoperação, havendo alternativas tais como Cython, Swig ou cffi. A biblioteca Boost do C++ inclui uma biblioteca para permitir a interoperabilidade entre as duas linguagens, e pacotes científicos fazem uso de bibliotecas de alta performance numérica escritos em Fortran e mantidos há décadas.

PYTHON

Cyber Mind

Comentários

Python fornece duas alternativas para documentar o código. A primeira é o uso de comentários para indicar o que certo código faz. Comentários começam com # e são terminados pela quebra da linha. Não há suporte para comentários que se estendem por mais de uma linha; cada linha consecutiva de comentário deve indicar #. A segunda alternativa é o uso de cadeias de caractere, literais de texto inseridos no código sem atribuição. Cadeias de caracteres em Python são delimitadas por " ou ' para única linha e por """ ou ''' para múltiplas linhas. Entretanto, é convenção usar o métodos de múltiplas linhas em ambos os casos.

Diferente de comentários, a cadeias de caracteres usadas como documentação são objetos Python e fazem parte do código interpretado. Isso significa que um programa pode acessar sua própria documentação e manipular a informação. Há ferramentas que extraem automaticamente essa documentação para a geração da documentação de API a partir do código. Documentação através de cadeias de caracteres também pode ser acessada a partir do interpretador através da função help().

PYTHON

Cyber Mind

Plataformas disponíveis

A linguagem e seu interpretador estão disponíveis para as mais diversas plataformas, desde Unix (Linux, FreeBSD, Solaris, MacOS X, etc.), Windows, .NET, versões antigas de MacOS até consoles de jogos eletrônicos ou mesmo alguns celulares, como a série 60, N8xx(PyMaemo) da Nokia e palmtops.

Para algum sistema operacional não suportado, basta que exista um compilador C disponível e gerar o Python a partir do fonte. O código fonte é traduzido pelo interpretador para o formato bytecode, que é multiplataforma e pode ser executado e distribuído sem fonte original.

Implementações

A implementação original e mais conhecida do Python é o CPython, escrita em C e compatível com o padrão C89, sendo distribuída com uma grande biblioteca padrão escrita em um misto de Python e C. Esta implementação é suportada em diversas plataformas, incluindo Microsoft Windows e sistemas Unix-like modernos.

PYTHON

Cyber Mind

Stackless Python é uma variação do CPython que implementa microthreads (permitindo multitarefa sem o uso de threads), sendo suportada em quase todas as plataformas que a implementação original.

Existem também implementações para plataformas já existentes: Jython para a Plataforma Java e IronPython para .NET.

Em 2005 a Nokia lançou um interpretador Python para os telefones celulares S60, chamado PyS60. Essa versão inclui vários módulos das implementações tradicionais, mas também alguns módulos adicionais para a integração com o sistema operacional Symbian. Uma implementação para Palm pode ser encontrada no Pippy. Já o PyPy, é a linguagem Python totalmente escrita em Python.

Diversas implementações, como CPython, pode funcionar como um interpretador de comandos em que o usuário executa as instruções sequencialmente, recebendo o resultado automaticamente. A execução compilada do código oferece um ganho substancial em velocidade, com o custo da perda da interatividade.

PYTHON

Cyber Mind

Desenvolvimento

O desenvolvimento de Python é conduzido amplamente através do processo Python Enhancement Proposal ("PEP"), em português Proposta de Melhoria do Python. Os PEPs são documentos de projeto padronizados que fornecem informações gerais relacionadas ao Python, incluindo propostas, descrições, justificativas de projeto (design rationales) e explicações para características da linguagem. PEPs pendentes são revisados e comentados por Van Rossum, o Benevolent Dictator for Life (líder arquiteto da linguagem) do projeto Python. Desenvolvedores do CPython também se comunicam através de uma lista de discussão, python-dev, que é o fórum principal para discussão sobre o desenvolvimento da linguagem. Questões específicas são discutidas no gerenciador de erros Roundup mantido em python.org. O desenvolvimento acontece no auto-hospedado svn.python.org

Licença

Python possui uma licença livre aprovada pela OSI e compatível com a GPL, porém menos restritiva. Ela prevê (entre outras coisas) que binários da linguagem sejam distribuídos sem a necessidade de fornecer o código fonte junto.

PYTHON

Cyber Mind

Módulos e frameworks

Ao longo do tempo têm sido desenvolvidos pela comunidade de programadores muitas bibliotecas de funções especializadas (módulos) que permitem expandir as capacidades base da linguagem. Entre estes módulos especializados destacam-se:

Descrição	Campos de atuação
Django	Framework para desenvolvimento ágil de aplicações web; desenvolvimento web
Pylons	Framework para desenvolvimento de aplicações web; desenvolvimento web
TurboGears	Framework baseado em várias outras tecnologias existentes no mundo que gira em torno da linguagem Python; desenvolvimento web
Matplotlib - Matplotlib / Pylab	biblioteca para manipulação de gráficos 2D; processamento de imagem
Python Imaging Library	biblioteca para manipulação de imagens digitais; processamento de imagem
PyOpenGL - Python OpenGL Binding	suporte multiplataforma ao OpenGL; computação gráfica

PYTHON

Cyber Mind

Pygame Conjunto de módulos para o desenvolvimento de jogos eletrônicos, incluindo gráficos SDL; desenvolvimento de jogos eletrônicos; computação gráfica

Twisted Framework para o desenvolvimento de aplicações de rede. Inclui módulos para servidor web, de aplicação, SSH e diversos outros protocolos; desenvolvimento de software; desenvolvimento web

PYRO - Python Remote Objects Framework para o desenvolvimento de sistemas distribuídos; computação distribuída

ZODB Sistema de persistência e banco de dados orientado a objetos; banco de dados

Plone SGC - Sistema de gerenciamento de conteúdo; desenvolvimento web

CherryPy Framework para aplicações web; desenvolvimento web

Web2py Framework para aplicações web; desenvolvimento web

Visual Python Framework 3D de alto nível; computação gráfica

SQLObject Mapeador objeto-relacional: traduz estruturas relacionais para objetos Python e manipula o banco de dados de forma transparente; banco de dados

Numarray Módulo para manipulação de vetores e computação científica. computação científica

PYTHON

Cyber Mind

Interfaces gráficas

Exemplos de bibliotecas de GUI disponíveis para Python incluem:

Descrição	
Tkinter	Módulo padrão para GUI no Python
PyGTK	interface para a biblioteca GTK+
PyQt	interface para a biblioteca Qt
wxPython	interface para a biblioteca wxWidgets
Etk	interface para a biblioteca EFL
Wax	Construído para simplificar o uso do wxPython
Kivy	Toolkit multiplataforma

Ambientes de desenvolvimento integrado

Existem vários ambientes de desenvolvimento integrado (IDE) disponíveis para Python:

PYTHON

Cyber Mind

IDE	Desenvolvedor	Última versão	Plataforma	Toolkit	Licença
IDLE	Guido van Rossum et al.	Distribuído com CPython	Multiplataforma	Tkinter	PSFL
PyCharm	JetBrains	2017.3 (29/11/2017)	Java	Swing	Apache 2.0
Komodo Edit	ActiveState	10 (17/05/2016)	Windows, Linux, macOS	XUL	MPL 1.1
Atom	GitHub	1.25.0 (15/03/2018)	Windows, Linux, macOS	Electron	MIT
GNOME Builder	GNOME Project	3.36.0 (06/03/2020)	Linux	GTK	GNU GPLv3+
Pyzo	Pyzo team	4.4.3 (09/10/2017)	Multiplataforma	PyQt	BSD
Boa Constructor	Team	0.6.1	Independente	wxPython	GNU GPL
Eric Python IDE	Detlev Offenbach	4.1.2	Independente	Qt	GNU GPL
Geany	Team	1.23	Independente	GTK2	GNU GPL
IronPython Studio	Clarius Labs	1.0 (10/122007)	Windows	VS2008 Shell Runtime	Microsoft Public License
PyDev (Eclipse)	Appcelerator	5.7.0 (11/04/2017)	Java	SWT	EPL
PythonCard	Alex Tweedly	0.8.2	Multiplataforma	wxPython	BSD
PyScripter	mmm-experts	1.7.2 (10/2006)	Windows	MIT	
Stani's Python Editor	Stani	0.8.4c (14/02/2008)	Independente	wxPython	GNU GPL
Spyder	Spyder developer community	2.3.2 (03/12/2014)	Windows, Linux, macOS	PyQt	MIT
Wing IDE	Wingware	3.0.2-1 (27/11/2007)	Windows, Linux, macOS	PyGTK	Proprietário

PYTHON

Cyber Mind

Aplicações

Alguns dos maiores projetos que utilizam Python são o servidor de aplicação Zope, o compartilhador de arquivos Mnet, o sítio YouTube e o cliente original do BitTorrent. Grandes organizações que usam a linguagem incluem Google[35] (parte dos crawlers), Yahoo! (para o sítio de grupos de usuários) e NASA. O sistema de gerenciamento de reservas da Air Canada também usa Python em alguns de seus componentes.[37] A linguagem também tem bastante uso na indústria da segurança da informação.

A linguagem tem sido embarcada como linguagem de script em diversos softwares, como em programas de edição tridimensional como Maya, Autodesk Softimage, TrueSpace e Blender. Programas de edição de imagem também a usam para scripts, como o GIMP.[40] Para diversos sistema operacionais a linguagem já é um componente padrão, estando disponível em diversas distribuições Linux. O Red Hat Linux usa Python para instalação, configuração e gerenciamento de pacotes.

Outros exemplos incluem o Plone, sistema de gerenciamento de conteúdo desenvolvido em Python e Zope e a Industrial Light & Magic, que produz filmes da série Star Wars usando extensivamente Python para a computação gráfica nos processos de produção dos filmes.

CSS

Cyber Mind

Cascading Style Sheets (CSS) é um mecanismo para adicionar estilo (cores, fontes, espaçamento, etc.) a um documento web.

O código CSS pode ser aplicado diretamente nas tags ou ficar contido dentro das tags <style>. Também é possível, em vez de colocar a formatação dentro do documento, criar um link para um arquivo CSS que contém os estilos. Assim, quando se quiser alterar a aparência dos documentos vinculados a este arquivo CSS, basta modifica-lo.

Com a variação de atualizações dos navegadores, o suporte ao CSS pode variar. A interpretação dos navegadores pode ser avaliada com o teste Acid2, que se tornou uma forma base de revelar quão eficiente é o suporte de CSS, fazendo com que a nova versão em desenvolvimento do Firefox seja totalmente compatível a ele, assim como o Opera já é. O Doctype informado, ou a ausência dele, determina o quirks mode ou o strict mode, modificando o modo como o CSS é interpretado e a página desenhada.

CSS

Cyber Mind

Sintaxe

Resultado obtido no Acid2 satisfatoriamente.

CSS tem uma sintaxe simples, e utiliza uma série de palavras em inglês para especificar os nomes de diferentes propriedade de estilo de uma página.

Uma instrução CSS consiste em um seletor e um bloco de declaração. Cada declaração contém uma propriedade e um valor, separados por dois pontos (:). Cada declaração é separada por ponto e vírgula (;).[2]

Em CSS, seletores são usados para declarar a quais elementos de marcação um estilo se aplica, uma espécie de expressão correspondente. Os seletores podem ser aplicados a todos os elementos de um tipo específico, ou apenas aqueles elementos que correspondam a um determinado atributo; elementos podem ser combinados, dependendo de como eles são colocados em relação uns aos outros no código de marcação, ou como eles estão aninhados dentro do objeto de documento modelo.

CSS

Cyber Mind

Pseudoclasse é outra forma de especificação usada em CSS para identificar os elementos de marcação, e, em alguns casos, ações específicas de usuário para o qual um bloco de declaração especial se aplica. Um exemplo frequentemente utilizado é o :hover pseudoclasse que se aplica um estilo apenas quando o usuário 'aponta para' o elemento visível, normalmente, mantendo o cursor do mouse sobre ele. Isto é anexado a um seletor como em a:hover ou #elementid:hover. Outras pseudoclasses e pseudoelementos são, p. ex., :first-line, :visited ou :before. Uma pseudoclasse especial é :lang(c), "c".

Uma pseudoclasse seleciona elementos inteiros, tais como :link ou :visited, considerando que um pseudoelemento faz uma seleção que pode ser constituída por elementos parciais, tais como :first-line ou :first-letter.

Seletores podem ser combinados de outras formas também, especialmente em CSS 2.1, para alcançar uma maior especificidade e flexibilidade.

CSS

Cyber Mind

Aqui está um exemplo que resume as regras acima:

```
selector [, selector2, ...][:pseudo-class] {
  property: value;
 [property2: value2;
  ...]
}
/* comment */
```

Seletores

Definição de estilo é um conjunto de propriedades visuais para um elemento, o CSS define regras que fazem as definições de estilo casarem com um elemento ou grupo de elemento, o documento pode conter um bloco de CSS num elemento style ou usando o elemento link apontando para um arquivo externo que contenha o bloco CSS.

Para uso com o CSS, foi criado o atributo class que todo elemento pode conter.

CSS

Cyber Mind

As regras de casamento para o CSS são chamadas de seletores, uma definição de estilo pode ser casada com um seletor ou um grupo de seletores separados por vírgula, um seletor pode casar um elemento por:

elemento do tipo : element_name { style definition; }

elemento do tipo com a classe : element_name.class_name { style definition; }

todos os elementos com a classe : .class_name { style definition; }

o elemento com o id : #id_of_element { style definition; }

casamento de um grupo : element_name_01, element_name_02, .class_name { style definition; }

Exemplos

```
p {text-align: right; color: #BA2;}
p.minhaclasse01 { color:#ABC; }
.minhaclasse02 { color:#CAD; }
# iddomeuelemento { color:#ACD; }
p.minhaclasse03 .minhaclasse04 { color:#ACD; }
```

CSS

Cyber Mind

Exemplos

```
p {text-align: right; color: #BA2;}
p.minhaclasse01 { color:#ABC; }
.minhaclasse02 { color:#CAD; }
# iddomeuelemento { color:#ACD; }
p.minhaclasse03 .minhaclasse04 { color:#ACD; }
```

Exemplos

```
/* comentário em css, semelhante aos da linguagem c */
body
{
   font-family: Arial, Verdana, sans-serif;
   background-color: #FFF;
   margin: 5px 10px;
}
```

CSS

Cyber Mind

O código acima define fonte padrão Arial, caso não exista, substitui por Verdana, caso não exista, define qualquer fonte sans-serif. Define também a cor de fundo do corpo da página.

Sua necessidade adveio do fato do HTML, aos poucos, ter deixado de ser usado apenas para criação de conteúdo na web, e portanto havia uma mistura de formatação e conteúdo textual dentro do código de uma mesma página. Contudo, na criação de um grande portal, fica quase impossível manter uma identidade visual, bem como a produtividade do desenvolvedor. É nesse ponto que entra o CSS.

As especificações do CSS podem ser obtidas no site da W3C "World Wide Web Consortium", um consórcio de diversas empresas que buscam estabelecer padrões para a Internet.

É importante notar que nenhum navegador suporta igualmente as definições do CSS. Desta forma, o web designer deve sempre testar suas folhas de estilo em navegadores de vários fabricantes, e preferencialmente em mais de uma versão, para se certificar de que o que foi codificado realmente seja apresentado da forma desejada.

HTML

Cyber Mind

HTML (abreviação para a expressão inglesa HyperText Markup Language, que significa Linguagem de Marcação de Hipertexto) é uma linguagem de marcação utilizada na construção de páginas na Web. Documentos HTML podem ser interpretados por navegadores. A tecnologia é fruto da junção entre os padrões HyTime e SGML.

HyTime é um padrão para a representação estruturada de hipermídia e conteúdo baseado em tempo. Um documento é visto como um conjunto de eventos concorrentes dependentes de tempo (como áudio, vídeo, etc.), conectados por hiperligações (em inglês: hyperlink e link). O padrão é independente de outros padrões de processamento de texto em geral.

SGML é um padrão de formatação de textos. Não foi desenvolvido para hipertexto, mas tornou-se conveniente para transformar documentos em hiper-objetos e para descrever as ligações.

Tabela de cores

Todo documento HTML possui marcadores (do inglês: tags), palavras entre parênteses angulares (chevron) (< e >); esses marcadores são os comandos de formatação da linguagem. Um elemento é formado por um nome de marcador (tag), atributos, valores e filhos (que podem ser outros elementos ou texto). Os atributos modificam os resultados padrões dos elementos e os valores caracterizam essa mudança. Exemplo de um elemento simples (não possui filhos):

HTML

<hr />

Exemplo de um elemento composto (possui filhos):

Wikipédia

<a> é o marcador de abertura

 é o marcador de fechamento

href é o atributo onde é definido a url, que será acessada ao clicar no link.

Outro exemplo de elemento composto (possui filhos):

 Wikipédia

p = marcador que define um parágrafo.

a = marcador que define uma hiperligação.

href = atributo que define a url da hiperligação.

target = atributo que define a forma como a hiperligação será aberta.

_self = valor do atributo Target que define que a hiperligação será aberta na mesma guia.

/ = define o fechamento do elemento

HTML

Cyber Mind

Isso é necessário porque os marcadores servem para definir a formatação de uma porção do documento, e assim marcamos onde começa e termina o conteúdo que receberá a formatação ou marcação necessária, específica. Alguns elementos são chamados “vazios”, pois não marcam uma região de texto, apenas inserem algum elemento no documento.

Cada elemento tem os seus atributos possíveis e os seus valores. Um exemplo, é o atributo href que pode ser usado com o elemento a, com o link mas que não pode ser usado com o elemento meta. Isso quer dizer que devemos saber exatamente quais os atributos e valores possíveis para cada elemento.

De uma maneira geral o HTML é um poderoso recurso, sendo uma linguagem de marcação muito simples e acessível voltada para a produção e compartilhamento de documentos, imagens, vídeos e áudio via streaming.

Na sua versão mais recente, o HTML5, é possível criar marcadores personalizados com JavaScript, linguagem de programação diretamente compatível com o HTML. Cada tag pode ter uma função específica utilizando uma API (Interface de programação de aplicações) diferente, assim como seus nomes e estilos.

HTML

Cyber Mind

Edição de documentos HTML

Os documentos em HTML são arquivos de texto simples que podem ser criados e editados em qualquer editor de textos comum, como o Bloco de Notas do Windows, ou o TextEdit, do Macintosh. Para facilitar a produção de documentos, no mercado existem editores HTML específicos, com recursos sofisticados, que facilitam a realização de tarefas repetitivas, inserção de objetos, elaboração de tabelas e outros recursos. Basicamente dividem-se em dois tipos:

- **Editores de texto fonte:** inserem automaticamente os marcadores, orientando a inserção de atributos e marcações
- **Editores WYSIWYG:** oferecem ambiente de edição com um "esboço" resultado final das marcações

Estrutura básica de um documento

A estrutura básica de um documento HTML (Hyper Text Markup Language - Linguagem de Marcação de Hypertexto), apresenta as seguintes marcações:

HTML

Cyber Mind

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8" />
        <meta name="description" content="a descrição do seu site em no máximo 90 caracteres">
        <meta name="keywords" content="escreva palavras-chaves curtas, máximo 150 caracteres">
        <title>Título do Documento</title>
    </head>
    <body>
    <!-- Aqui fica a página que será visível para todos, onde pode-se inserir
    textos, imagens, links para outras páginas, etc, geralmente usa-se: -->

    <div>Tag para criar-se uma 'caixa', um bloco, mais utilizada com "Cascading Style Sheets
        (Folhas de Estilo em Cascata)</div>

    <span>Tag para modificação de uma parte do texto da página</span>

    <img src="endereço_de_uma_imagem.jpg" />

    <a href="http://www.wikipedia.org">Wikipedia, A Enciclopédia Livre</a>
</body>
</html>
```

Os marcadores HTML não são sensíveis à caixa, portanto tanto faz escrever <HTML>, <Html>, <html> ou <HtMl>.

HTML

Cyber Mind

Os marcadores básicos de HTML, cuja presença é altamente recomendada nas páginas são:

<html>: define o início de um documento HTML e indica ao navegador que todo conteúdo posterior deve ser tratado como uma série de códigos HTML

<head>: define o cabeçalho de um documento HTML, que traz informações sobre o documento que está sendo aberto

<body>: define o conteúdo principal, o corpo do documento. Esta é a parte do documento HTML que é exibida no navegador. No corpo podem-se definir atributos comuns a toda a página, como cor de fundo, margens, e outras formatações.

Cabeçalho

Dentro do cabeçalho podemos encontrar os seguintes elementos:

<title>: define o título da página, que é exibido na barra de título dos navegadores

<style type="text/css">: define formatação em CSS

<script type="text/javascript">: define programação de certas funções em página com scripts, podendo adicionar funções de JavaScript

HTML

Cyber Mind

<link>: define ligações da página com outros arquivos como feeds, CSS, scripts, etc

<meta>: define propriedades da página, como codificação de caracteres, descrição da página, autor, etc

São meta informações sobre documento. Tais campos são muito usados por mecanismos de busca (como o Google, Yahoo!, Bing) para obterem mais informações sobre o documento, a fim de classificá-lo melhor. Por exemplo, pode-se adicionar o código <meta name="description" content="descrição da sua página" /> no documento HTML para indicar ao motor de busca que texto de descrição apresentar junto com a ligação para o documento. Para o motor de busca Google, por exemplo, elementos meta como keywords não são utilizadas para indexar páginas. Apenas <title> e a meta <description> são usadas para descrever a página indexada.

Corpo

Trecho de código HTML.

Dentro do corpo podemos encontrar outros vários marcadores que irão moldar a página, como por exemplo:

<h1>, <h2>, ... <h6>: Títulos que variam de tamanho dentro das prioridades (sua aparência pode ser alterada com CSS - Cascade Style Sheet - Folhas de Estilo em Cascata).

<p>: Parágrafo.

HTML

: quebra de linha.

<table>: cria uma tabela (linhas são criadas com <TR> e novas células com <TD>, já os cabeçalhos das colunas são criados com os marcadores <THead><TH> e os rodapés com <TFooter><TR><TD>).

: determina uma divisão na página a qual pode possuir variadas formatações.

<b>, <i>, <u> e <s>: negrito, itálico, sublinhado e riscado, respectivamente.

<img />: imagem.

<a>: hiper-ligação para um outro local, seja uma página, um e-mail ou outro serviço.

<textarea>: caixa de texto (com mais de uma linha); estas caixas de texto são muito usadas em blogs, elas podem ser auto selecionáveis e conter outros códigos a serem distribuídos.

<abbr>: abreviação (sigla simplesmente abreviada).

<cite>: citação.

<address>:Endereço.

Cores

As cores devem ser declaradas em CSS, com o atributo style, que funciona em diversos elementos, como por exemplo:

HTML

Texto

Onde COR pode ser o nome da cor em inglês, em decimal, hexadecimal, RGB, RGBA ou HSLA. Exemplos: Tabela de cores e Lista de cores.

Hiperligações

Uma possibilidade importante dos documentos HTML é a de fazer hiperligações. Para isso usa-se o marcador <a> (do inglês, anchor). Esta tem os atributos: href que define o alvo da hiperligação (que pode ser uma página de Internet, uma parte da mesma página ou um endereço de email) ou name que define um alvo nessa página (a onde se pode fazer uma hiperligação usando o marcador a com o atributo href). Exemplos:

Clique aqui para aceder à página principal da Wikipédia em português.

texto

Em que nome e texto podem ser substituídos por o que se desejar. Depois usa-se   para hiperligar a este "anchor".

HTML

Cyber Mind

Diferença entre target="_blank" e target="_new"

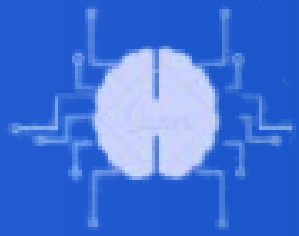
target="_blank" é usado para abrir links em várias janelas e target="_new" ou target="booger" é usado para abrir vários links em uma janela.[8]

Exemplos

Título
Título
Título
Página em branco é usado about:blank na url do link.

Exemplos:

Página em branco
Página em branco
Página em branco
Caracteres especiais e símbolos
Os caracteres especiais definem-se usando comandos que começam com & e terminam com um ;. Alguns exemplos incluem á (á), à (à), ã (ã), â (â), ä (ä) e ç (ç). Qualquer outra vogal pode ser substituída pelo a destes exemplos, incluindo maiúsculas.

Cyber Mind 

Parabéns! Você chegou ao fim do material.

A CyberMind agradece a confiança e esperamos que todo o conhecimento adquirido seja aplicado em sua vida profissional. Sucessos!