

Aula 0

Apresentação do Curso

Responsável

Prof. Armando Toda (armando.toda@usp.br)

Informações sobre o curso

- Prof. Armando Toda - sala 4-234 –
armando.toda@usp.br
- Site do curso
 - e-disciplinas
 - Codebench
- Material
 - Todo o material apresentado em aula será disponibilizado no site

Objetivos do Curso

Aprender

Programação

Objetivo do Curso

Traduzir os conceitos de Introdução a Ciência da Computação para linguagens de programação

- Linguagem do curso: C

ATENÇÃO

- Para quem já sabia linguagem de programação
 - Não adianta apenas saber programar
 - É preciso **pensar sobre e criar algoritmos**
- Para quem não sabia programar
 - Ajuda a pensar e desenvolver habilidades para criar algoritmos

Oportunidades

- Ajude seus colegas a estudar (não a colar)
- Faça network
- Aproveite as bolsas de estudo
- Vá para exterior com bolsa
- Dê sempre o seu melhor
- Extraia o máximo dos professores
 - Às vezes eles tem pouca didática, mas são muito bons nas áreas de pesquisa (INOVAÇÃO)

Vamos (tentar) evitar

- Pensamento 5.0 (cinco bola)
- Colar nas provas
- Chupinhar exercícios e projetos
- Assinar a lista de presença por outros
- Deixar outros assinem por você
- Desistir do curso
- Pedir “esmola” pro professor no final do semestre

Metodologia

- Resolução de exercícios
 - Listas de exercícios a serem realizadas em sala e no Codebench
- 2 ou 3 Provinhas em Grupos
 - Codebench
- 2 Provas e -possivelmente- um trabalho final
 - Individual.
 - No Codebench

Avaliação (Provisória)

- Média Exercícios em Sala (E)
- Média das Provas (P)
- Média das Provinhas (pv)
- Média Final (M)

Se $3 \leq (P \text{ ou } pv) \leq 5$ então

$$\mathbf{M = \min \{P, pv\}}$$

Senão

$$\mathbf{(pv * 0.3 + P * 0.7) + \text{Bonus (E)}}$$

Recuperação

- Se $M \geq 5$ e frequência $\geq 70\%$: **APROVADO**
- Se $3 \leq M < 5$ e frequência $\geq 70\%$ **RECUPERAÇÃO**
- Se $M < 3$ **ou** frequência $< 70\%$ **REPROVADO**
- Recuperação (Nrec)
 - $Nrec \geq 5 \Rightarrow M = 5$ **APROVADO**
 - $Nrec < 5 \Rightarrow M = \max \{ NP, Mrec \}$ **REPROVADO**

Presença e Prova Substitutiva

- O controle de presença é um requisito imposto pela USP, assim o controle será feito por meio de listas
- NÃO haverá prova substitutiva (sub)
 - Caso falte uma aula, é preciso conversar com o professor para pedir um exercício/prova extra

- Juiz Online desenvolvido pela UFAM
<https://codebench.icomp.ufam.edu.br/>

Que tal um sistema

JUIZ ONLINE

para correção automática dos códigos de seus alunos?

- Juiz Online desenvolvido pela UFAM
<https://codebench.icomp.ufam.edu.br/>



Matrícula em Novas Turmas

Na listagem abaixo, procure pela disciplina e turma em que você deseja se matricular no sistema CodeBench. É importante identificar corretamente a turma, e para isso basta identificar seu professor e os horários de aula. Uma vez identificada a turma, clique no botão que se encontra abaixo da opção **Matricular-se**. Você verá que botão mudará para a cor azul, indicando que sua matrícula foi feita com sucesso.



CODEBENCH

- Juiz Online desenvolvido pela UFAM
- <https://codebench.icomp.ufam.edu.br/>



CODEBENCH

HOME TURMAS ▾ IDE ▾ SOBRE ▾  ARMANDO ▾

[UFPR] Python para quem está começando (Escola de Verão F-P&C)

Matricular-se Professor / Turma

Semestre acessá-la. **Matrícula feita com sucesso:** Clique no nome da turma para



Heleno de Souza Campos Junior / Turma 1

2022/2 — — — — —

[USP] Laboratório de Introdução à Ciência de Computação I

Matricular-se Professor / Turma

Semestre seg ter qua qui sex



Armando Toda / Turma 1

Acessar

2022/2 — — — — — 08-10

[UTFPR] Computação I

- Juiz Online desenvolvido pela UFAM

<https://codebench.icomp.ufam.edu.br/>



HOME TURMAS ▾ IDE ▾ SOBRE ▾ ARMANDO ▾

Professor(a) Armando Toda — Turma 1



Home > Turma 1

[USP] Laboratório de Introdução à Ciência de Computação I

Início

Trabalhos 2

Materiais

Mensagens

Cartas

Gamificação



Armando Toda
PROFESSOR(A)

Objetivos da disciplina

Definir objetivos

Emblemas da gamificação

Você pertence ao grupo Bronze, representado pelo emblema abaixo. Existem três grupos: Ouro, Prata e Bronze. O seu grupo é determinado pela quantidade de pontos de experiência (Exp) que você adquiriu dentro do ambiente da gamificação.



Os emblemas abaixo representam o seu desempenho nas atividades da disciplina de programação. Os emblemas podem ser de Ouro, Prata ou Bronze. O primeiro emblema, de Bronze, representa a sua média nas avaliações (0) feitas até então; o segundo, de Bronze, representa a sua média nas listas de exercícios (0); e o terceiro emblema, de Bronze, representa a frequência com que você acessa o CodeBench.



Progresso Individual

- Juiz Online desenvolvido pela UFAM

<https://codebench.icomp.ufam.edu.br/>



HOME TURMAS ▾ IDE ▾ SOBRE ▾ ARMANDO ▾

Professor(a) Armando Toda — Turma 1



Home > Turma 1 > Trabalhos

[USP] Laboratório de Introdução à Ciência de Computação I

Início

Trabalhos 2

Materiais

Mensagens

Cartas

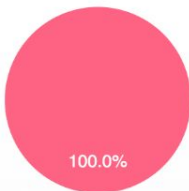
Gamificação

Trabalhos da turma

Mostrar Tudo

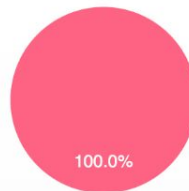
Laboratórios de Programação

Avaliações Práticas



2. ESTRUTURAS CONDICIONAIS

⌚ 31/03/2023 00:00 até 14/04/2023 00:00
em andamento



1. OPERAÇÕES BÁSICAS

⌚ 24/03/2023 00:00 até 07/04/2023 00:00
em andamento

- Juiz Online desenvolvido pela UFAM

<https://codebench.icomp.ufam.edu.br/>

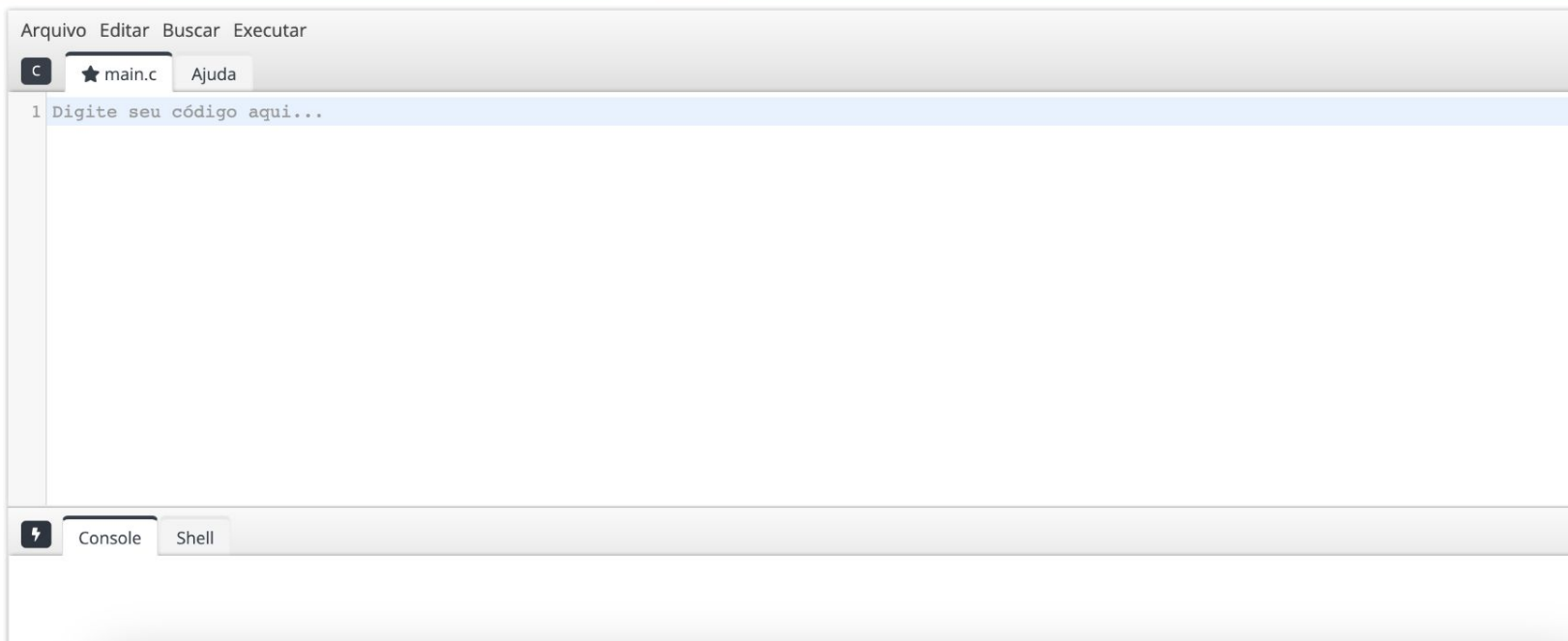


HOME TURMAS ▾ IDE ▾ SOBRE ▾ ARMANDO ▾

Linguagem C
Ambiente Integrado de Desenvolvimento

Home > Ambiente Integrado de Desenvolvimento

C C++ Python 3 Java Haskell Lua 5.3 Prolog Script Bash (com AWK e SED) Assembly (ARM)



Nosso primeiro programa

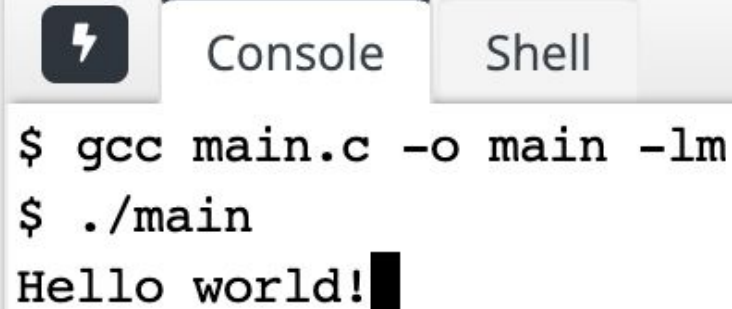
```
#include <stdio.h>

int main(void){
    printf("Hello world!");
    return 0;
}
```

Nosso primeiro programa

```
#include <stdio.h>

int main(void){
    printf("Hello world!");
    return 0;
}
```



Console Shell

```
$ gcc main.c -o main -lm
$ ./main
Hello world!
```

Tipos de variáveis

O que é uma variável?

É um espaço na memória do computador para armazenar um dado

É importante especificar o tipo de dado que aquela variável pertence

Tipos de variáveis

Tipos de dados no C

char: armazena um caractere, ex: a,b,c;

int: armazena um número inteiro, ex:
1,2,3;

float: armazena um número real com
certa precisão, ex: 1.5;

double: armazena um número real com
precisão maior que float, ex: 1.567654;

Tipos de variáveis

Tipo	Num de bits	Formato i/o	Início	Fim
char	8	%c	-128	127
unsigned char	8	%c	0	255
int	32	%d	-2.147.483.648	2.147.483.647
unsigned int	32	%u	0	4.294.967.295
long int	32	%li	-2.147.483.648	2.147.483.647
unsigned long int	32	%lu	0	4.294.967.295
short int	16	%hi	-32.768	32.767
unsigned short int	16	%hu	0	65.535
float	32	%f	$(+/-)10^{-38}$	$(+/-)10^{38}$
double	64	%lf	$(+/-)10^{-308}$	$(+/-)10^{308}$
long double	96			

Fonte: <https://www.ic.unicamp.br/~ducatte/mc102/aula03.pdf>

Tipos de variáveis

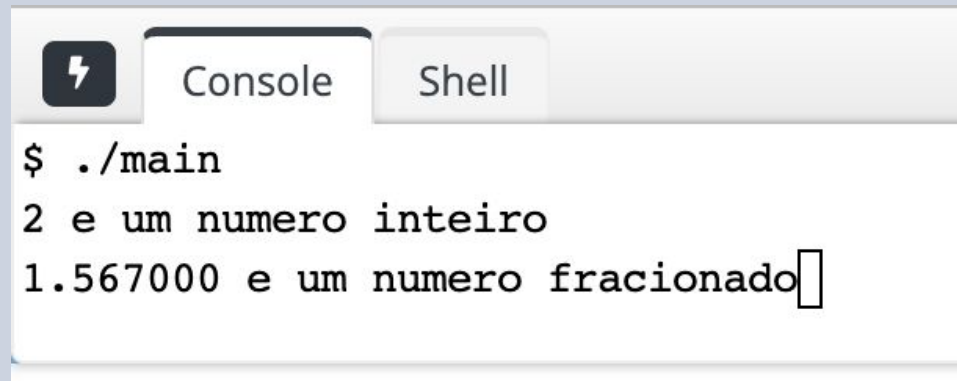
```
#include <stdio.h>
```

```
int main(void){  
    int a = 2;  
    float b = 1.567;  
    printf("%d e um numero inteiro\n%f e um numero fracionado", a, b);  
    return 0;  
}
```

Tipos de variáveis

```
#include <stdio.h>
```

```
int main(void){
    int a = 2;
    float b = 1.567;
    printf("%d e um numero inteiro\n%f e um numero fracionado", a, b);
    return 0;
}
```



```

$ ./main
2 e um numero inteiro
1.567000 e um numero fracionado

```