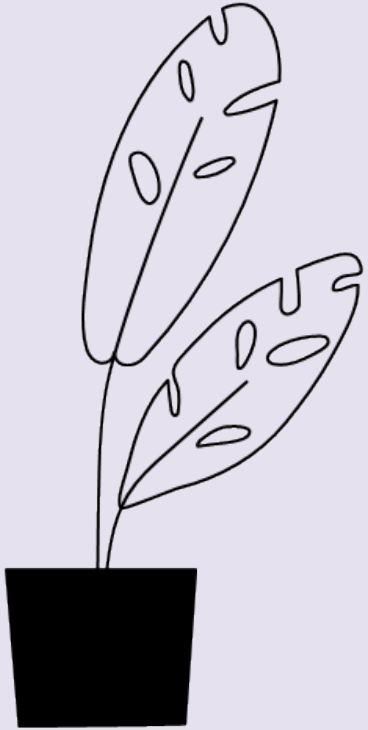


مفاهیم پایه زبان

مبانی کامپیوتر

استاد مصطفوی

ارائه دهندگان:
هلیا روزبهانی
فاطمه یحیوی
نیوشا شعاعی
محمد نایبی
حسام جعفری

1404/8/25

A

فهرست مطالب

A

معرفی زبان C و تاریخچه آن

B

و ساختار main ابتدایی برنامه

C

متغییرها ثابت‌ها و مقادیر در c

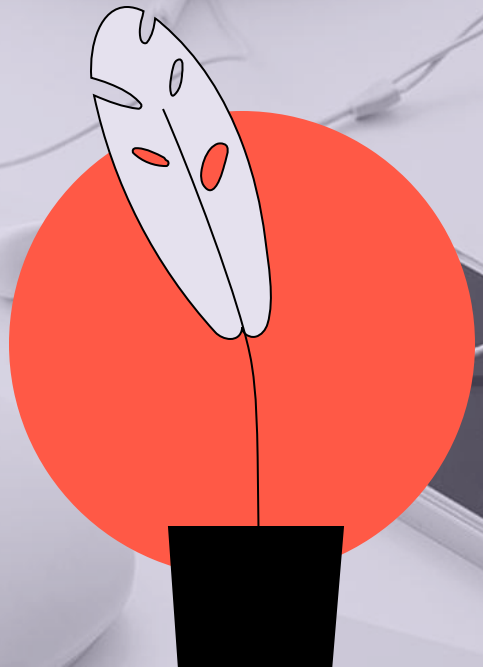
D

توسعه برنامه‌های مقدماتی: نوشتن و اجرای برنامه‌های ساده

E

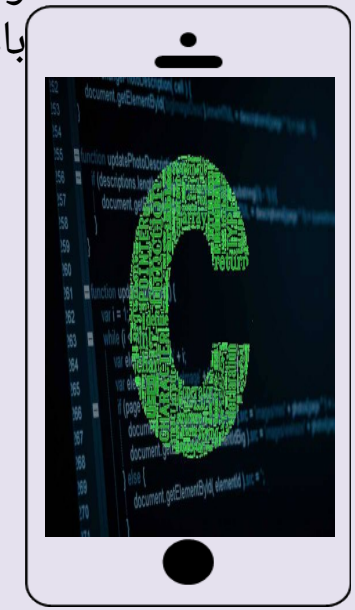
محاسبات ریاضی و قالب بندی ورودی و خروجی

بخش اول



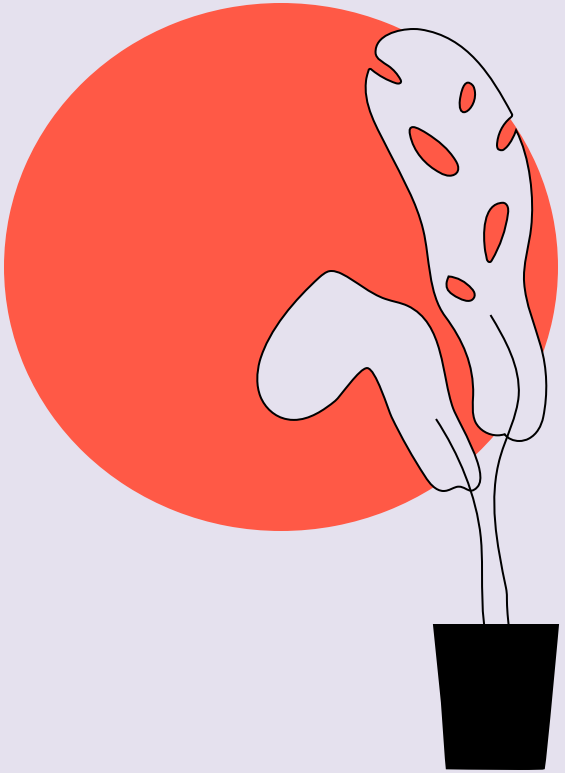
چرا c سی شد

زبان B قبل از C خلق و استفاده می شد
ولی کمبود امکانات B برای ساخت UNIX
باعث خلق زبانی قدرتمند و پیشرفته تر به
نام C شد .

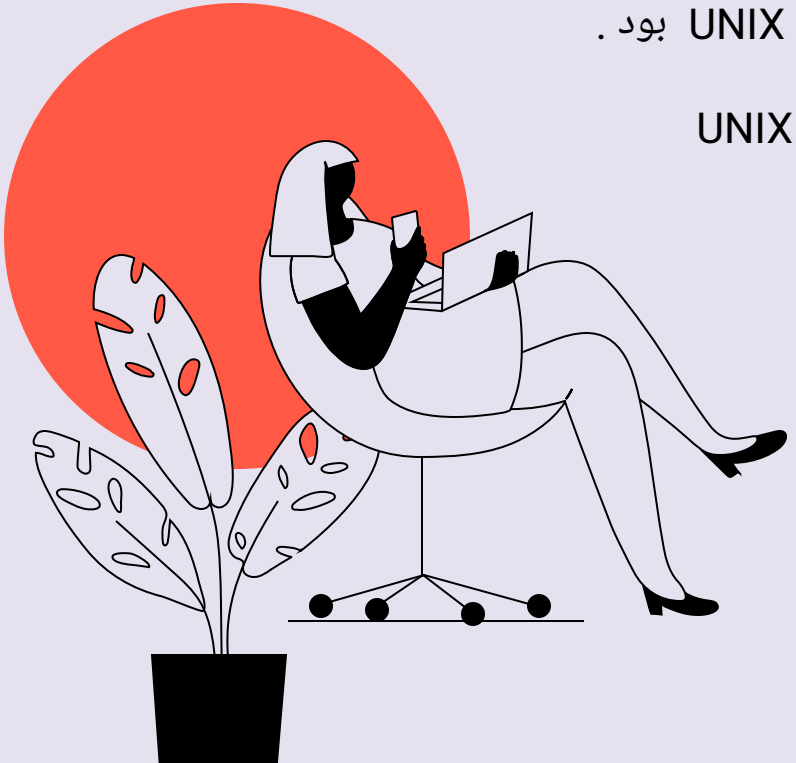


شناسنامه C

زبان C اوایل ۱۹۷۰ در ازمایشگاه بل توسط دنیس ریچی خلق شد .
زبان C از اسمبلی ساده تر بهینه تر و قابل حمل تر بود
ولی همزمان به سخت افزار هم نزدیک بود!!
منظور از قابل حمل بودن یعنی با یک تغییر کوچیک میتونست روی سیستم عامل های دیگر هم اجرا شود .



!! داداشی هم UNIX و C



نتیجه خوانایی و قابل حمل بودن زبان C به وجود آمدن سیستم عامل UNIX بود .
سی و UNIX با هم بزرگ شدن
سی به UNIX کمک کرد تا سیستم عاملی قوی و قابل حمل تبدیل بشه و UNIX
هم باعث شهرت و گسترش C شد .
و در نتیجه آن C به الگو پایه بسیاری از زبان ها تبدیل شد .

C + oop ---> C++
C / C++ ≈ java
حتی C# و python هم از C الهام گرفتند .

میگن »هرکس C را فراگیرد بقیه زبان هارا سهولت می‌آموزد. «

ویژگی های زبان C

نزدیکی به سخت افزار

میتوانیم مستقیما با حافظه و سخت افزار کار کنیم.
(ویژگی که تو خیلی از زبان های مدرن وجود ندارد)

سرعت بالا

مستقیما به زبان ماشین کامپایل می شود و به همین خاطر بشدت سریع است .

قابلیت حمل

با کمی تغییر روی Cکد های سیستم عامل ها و پردازنده های مختلف اجرا می شوند.

سادگی و خوانایی

برخلاف زبان های قبل از Cزبان خودش به زبان انگلیسی نزدیک بود و خوانایی بیشتری داشت.

مستقیما به زبان ماشین کامپایل
می شود و به همین خاطر
بشدت سریع است .

سرعت بالا

میتوانیم مستقیما با حافظه و سخت افزار کار کنیم.
)ویژگی که تو خیلی از زبان های مدرن وجود ندارد(

نزدیکی به سخت افزار

سادگی و خوانایی

زبان Cبرخلاف زبان های قبل از خودش به زبان انگلیسی نزدیک بود و خوانایی بیشتری داشت.

قابلیت حمل

کد های C با کمی تغییر روی سیستم عامل ها و پردازنده های مختلف اجرا می شوند.

نسخه های مختلف C

C98 / ANSI C
اولین نسخه رسمی و استاندارد که در سال ۱۹۸۹ معرفی شد.

C99
اضافه شدن دادهای جدید مثل long long int
اضافه شدن کتابخانه math.h و ثابت های عددی
توانایی تعریف متغیر ها در هر بلوک
امکان تعریف ارایه ها با اندازه متغیر

C11 و c18
با تغییرات بالاتر از سطح کلاس ما!

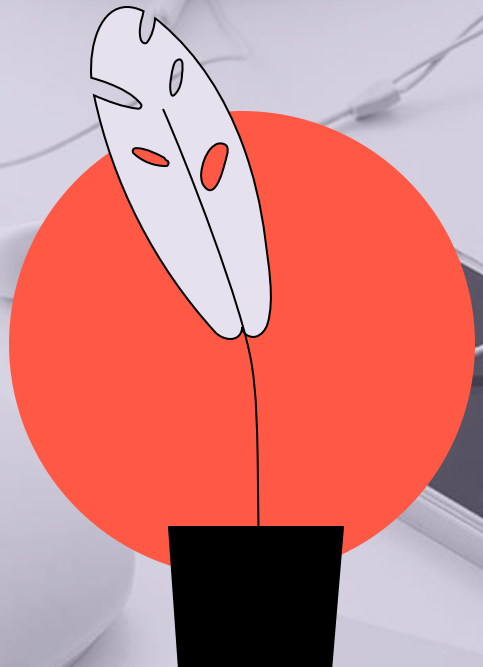
کاربردهای C

- سیستم عامل ها (Windows , Linux , UNIX , mcOS) که بخش زیادی از کد های آن ها را با C نوشتند .
- میکرو کنترلرها : برای کنترل سخت افزار وسایل الکترونیکی در لوازم خانگی خودرو و تجهیزات صنعتی از C استفاده میشود .
- نرم افزار های سیستمی مثل کامپایلر ها مفسر ها و درایور ها

به طور خلاصه

زبانی است که در قلب دنیای فناوری جا دارد بدون C آن شاید خیلی از سیستم عامل ها و زبان هایی که امروز می شناسیم وجود نداشتند .

بخش دوم



ساختار کلی یک برنامه

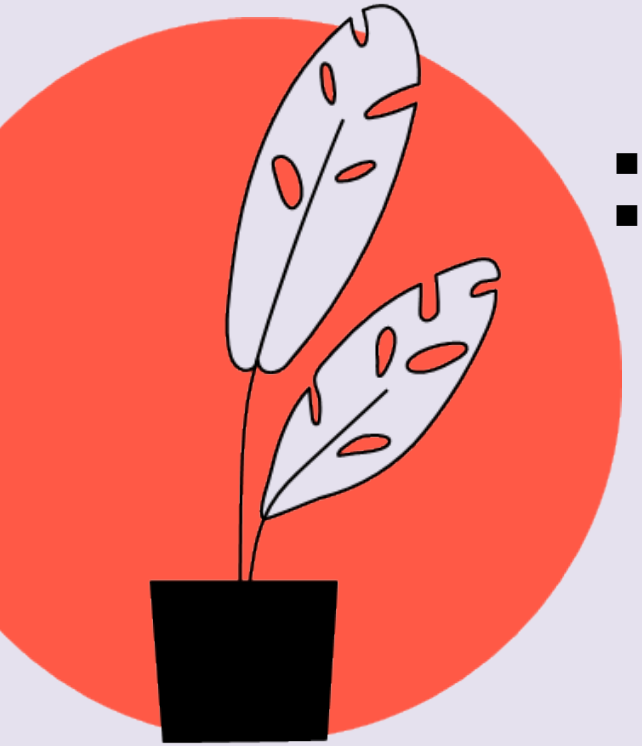
هر برنامه C از چند بخش اصلی تشکیل شده است:

Header files
با استفاده از دستور #include آن‌ها را به برنامه اضافه می‌کنیم
هدر فایل یا فایل های با پسوند .h نقشه راه برنامه های C هستند که به کامپایلر توضیح میدهد چه توابع و تعاریفی در کد اصلی ما به کار برده شده است
همچنین به ما اعلام میکنه که چه پارامتر هایی استفاده شدند .
برای مثال: string.h مربوط به کار با رشته ای از کاراکتر ها میشود.

ساختار اولیه و کلی یک برنامه C به صورت زیر می باشد:

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

تابع main



نقطه شروع اجرای برنامه است.

هر برنامه C باید دقیقا یک تابع main داشته باشد.

در پایان return تعیین می‌کند نتیجه اجرای برنامه چه بوده است:

- یعنی برنامه بدون خطا تمام شد.0 return
- یا هر عدد دیگر معمولا برای نمایش خطاها استفاده می‌شود. 1 return

```c
int main(){
    printf("Hello World");
    return 0;
}
```

نحوه ی استفاده از کتابخانه و تابع printf stdio.h

مخفف Standard Input/Output است. Stdio.h
این کتابخانه توابعی را برای ورودی و خروجی فراهم می‌کند
مثل:

- برای نمایش متن روی صفحه printf
- برای گرفتن ورودی از کاربر scanf

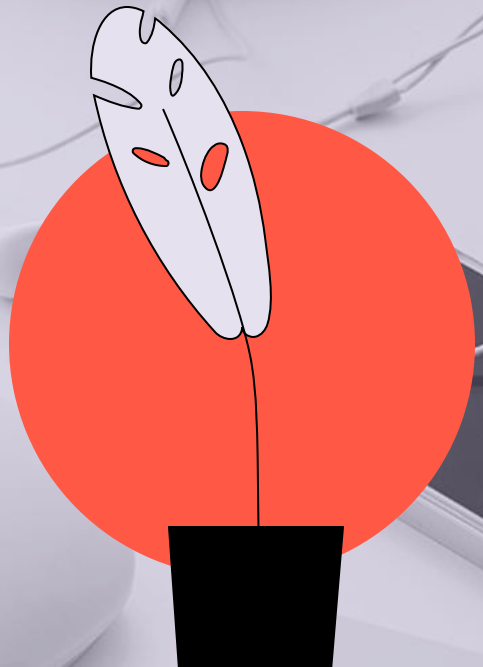
```c
#include <stdio.h>
int main(){
    printf("Be Zaban C Khosh Omdi!");
    return 0;
}
```

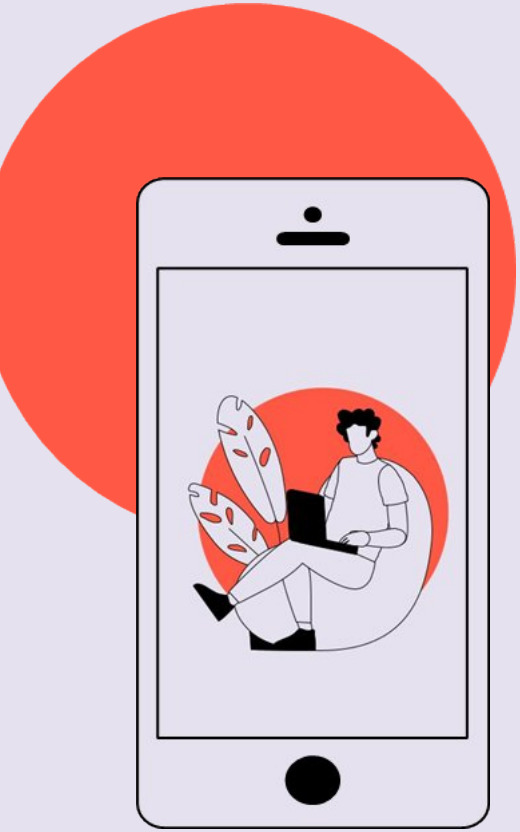
مفهوم کامپایل
اجرا و خطاهای زمان کامپایل

1. کامپایل Compile
- کد نوشته‌شده توسط کامپایلر به زبان ماشین)کد قابل اجرا(ترجمه می‌شود.
- در این مرحله اگر اشتباه‌های دستوری <Syntax Errors > وجود داشته باشد کامپایلر آن‌ها را گزارش می‌کند.

2. اجرا Run
- اگر کد با موفقیت کامپایل شد برنامه اجرایی ساخته می‌شود و می‌توان آن را اجرا کرد.

بخش سوم





متغیرها ثابت ها و مقادیر در c

تو هر زبان برنامه‌نویسی برای ذخیره‌ی اطلاعات باید از متغیر ها و ثابت ها استفاده کنیم در زبان c هم متغیر ها نقش ظرف هایی رو دارند که داده ها را در حافظه نگه میدارند.

با تعریف نوع داده)data type(مشخص می‌کنیم که چه نوع اطلاعاتی در این صفحه ذخیره میشود .

انواع داده

برای ذخیره یک کاراکتر مثل حرف یا عدد مثل:
char A4 $

برای ذخیره اعداد اعشاری با دقت پانزده عدد پشت اعشار
double

برای ذخیره اعداد اعشاری با دقت هفت عدد پشت اعشارfloat

برای ذخیره اعداد صحیح بدون اعشار
int

اصلاح کننده نوع داده

در زبان C می‌توان با کلمات کلیدی نوع داده ها را تغییر داد تا بازه یا اندازهی آنها تغییر کند.

کلماتی مثل :

Short, long

که برای ذخیره عدد های خیلی کوچکتر یا خیلی بزرگتر از محدوده بکار میروند و با آن ها میتوانیم اندازه وظرفیت را تغییر بدهیم.

Unsigned, signed

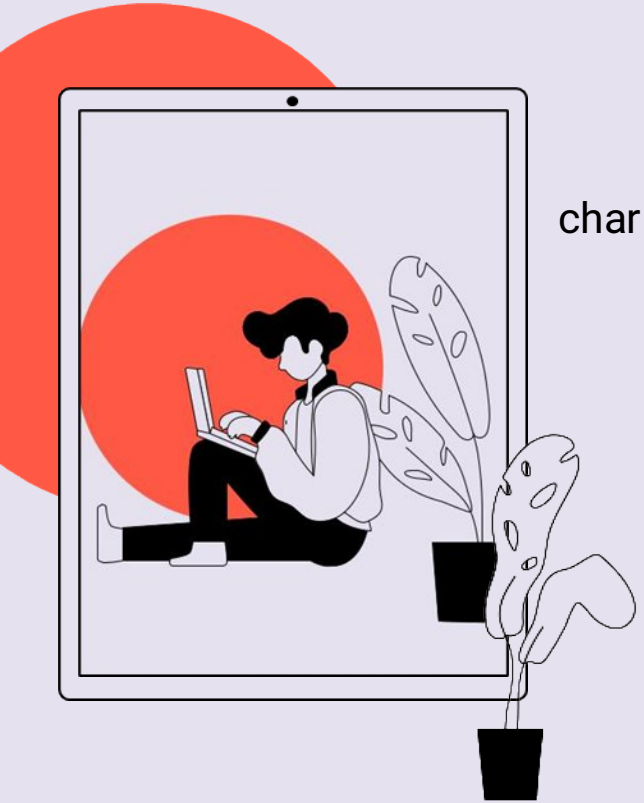
این دو برای تعیین علامت دار بودن یا نبودن داده بکار میروند.

حافظه اشغال شده توسط هر نوع داده

هر int 4 بایت اشغال می کند ولی short int 2 بایت و long long int 8 بایت.
ولی علامت دار بودن یا نبودن یک داده فرقی در حجم حافظه اشغالی ندارد چون هر و یک حجم از اعداد را در بر می گیرند.

C Basic Data Types	32-bit CPU		64-bit CPU	
	Size (bytes)	Range	Size (bytes)	Range
char	1	-128 to 127	1	-128 to 127
short	2	-32,768 to 32,767	2	-32,768 to 32,767
int	4	-2,147,483,648 to 2,147,483,647	4	-2,147,483,648 to 2,147,483,647
long	4	-2,147,483,648 to 2,147,483,647	8	-9,223,372,036,854,775,808-9,223,372,036,854,775,807
long long	8	9,223,372,036,854,775,808-9,223,372,036,854,775,807	8	9,223,372,036,854,775,808-9,223,372,036,854,775,807

توضیحChar و string



ما با استفاده از char میتوانیم یک کاراکتر ذخیره کنیم مثل @ # A 3
هر char دقیقا ۱ بایت فضا میگیرد.
نکته: برای مقدار دهی به charباید حتما از ' ' استفاده کرد.

char x = '$';

استرینگ)string(برای ذخیره نخ و رشته ای از کاراکتر ها که بهم وصل هستند
استفاده میشود در اخر متغیر باید [ ] بزاریم.

Char name[ ] = “newsha";

حجم استرینگ بستگی به طول آن دارد و متغیر است .

نکته: برای مقدار دهی به استرینگ از " " استفاده می کنیم.

مقدار دهی

برای تعریف متغیر و مقدار دهی به آن به روش زیر عمل میکنیم:

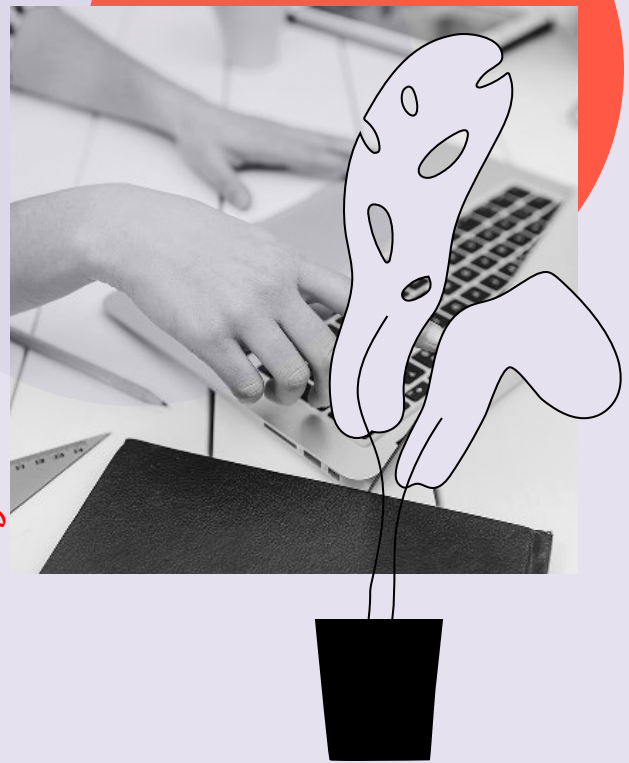
ابتدا نوع داده سپس امس متغیر و در اخر مقدار آن را مشخص میکنیم .

```
int x = 10;
char name[ ] = "heheheha";
```

یا میتوانیم به صورت جدا ابتدا متغیر را تعریف و سپس مقدار دهی کنیم .

```
float y;
y = 10.34;
```

هشدار: اگه متغیری تعریف کنیم ولی به ان مقداری ندیم مقدار تو حافظه تصادفی میشه پس بهتره یه مقدار اولیه بدیم .



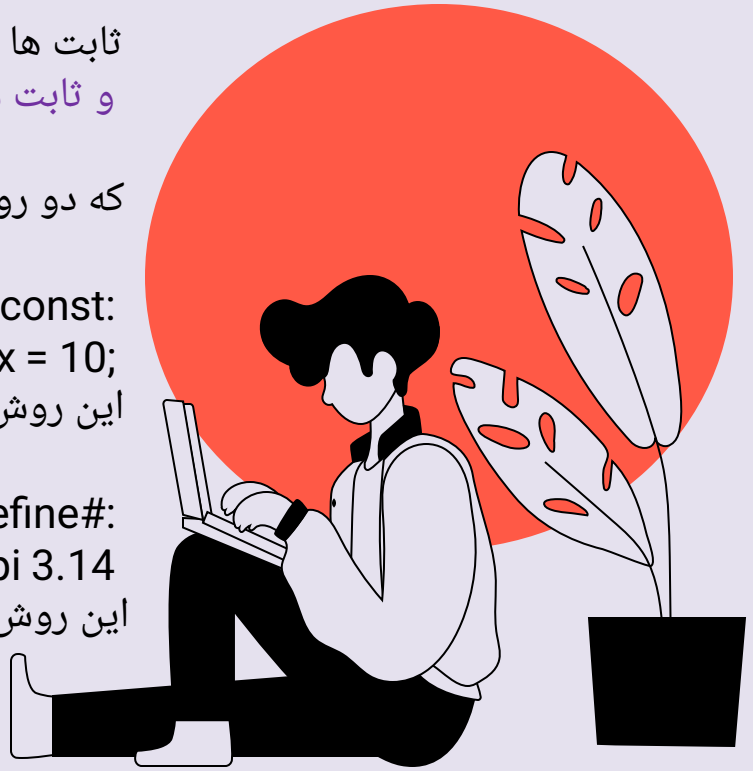
ثابت constant

ثابت ها دقیقا مانند متغیر ها مقدار میگیرند ولی مقدار انها در طول برنامه تغییر نمیکنه
و ثابت میمونه .

که دو روش برای تعریف ثابت وجود داره:

۱. استفاده از const:
const int x = 10;
این روش در زمان کامپایل کنترل میشه و قابل تغییر نیست .

۲.استفاده از define#:
#define pi 3.14
این روش در مرحله قبل از کامپایل جایگزین میشود .



قوانین نام گذاری متغیر ها

قوانین زبان برنامه نویسی:

- نام فقط می‌تواند شامل حروف اعداد و آندراسکور)_(باشد .
- نام نباید با عدد شروع شود .
- زبانC به حروف کوچک و بزرگ حساس است یعنی Age با age فرق داره .
- نباید از کلمات کلیدی مثلint, if, return استفاده شود .

قوانین عمومی بین برنامه نویس ها:

- برای مثال بهتره نام‌ها معنی‌دار باشد مثلا : first_name به‌جای x .

محدوده دید scope

یعنی هر متغیر در کدام بخش از برنامه قابل استفاده است .

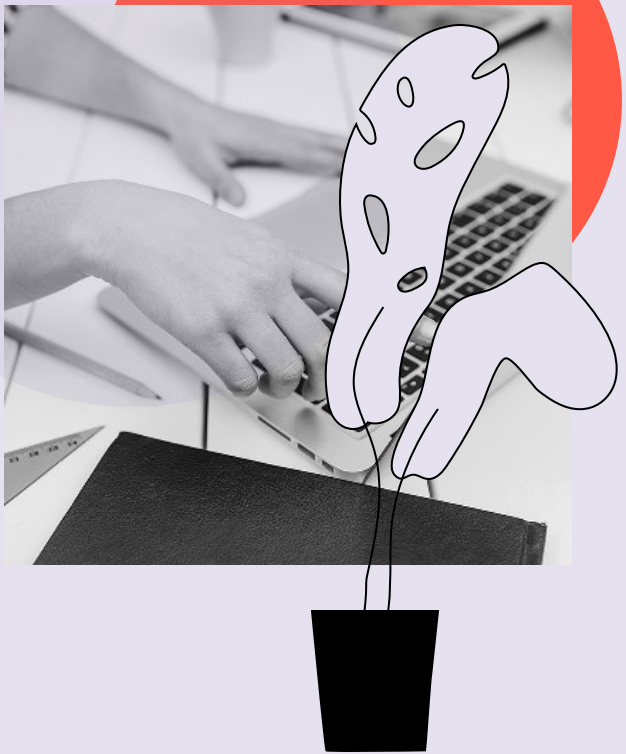
متغیر محلیlocal فقط در بلوک یا تابعی که تعریف شده میتوان استفاده شود . √

متغیر سراسری global در سرتاسر برنامه میتوان استفاده شود . √

```
// ;int a  = 5   global

// ;int b = 10   local
```

```
void func() {

}
```

تفاوت نوع داده های عددی و کاراکتری

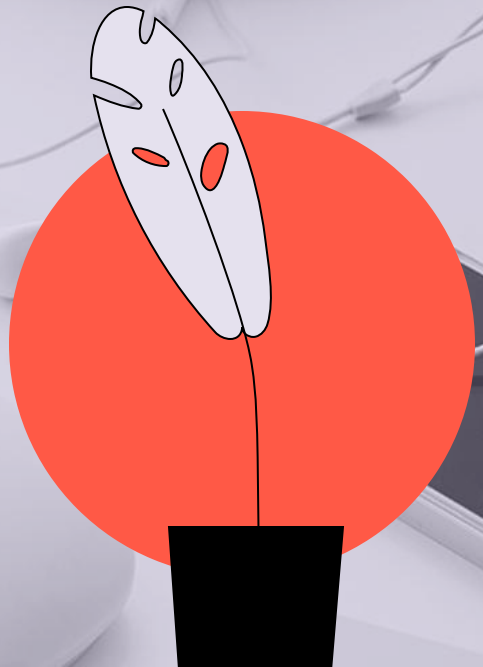


اینکه ما یک داده مثل ۵ را به صورت عدد یا کاراکتر ذخیره کنیم فرق زیادی دارد .

- بزرگترین فرق اینه اگه به صورت عدیint, float, double ذخیره شود توانایی انجام محاسبات ریاضی دارد .
- ولی به صورت کاراکترchar این توانایی رو ندارد .

- دومین فرق تو نوع ذخیره که int 5 4 بایت ولی کاراکتر '5' تنها ۱ بایت است .

بخش چهارم



در زبان Cبرای فرا خواندن متغیرها باید از مشخص کننده های فرمت
استفاده کنیم.(format specifier)
برای عدد صحیح %d
برای اعداد اعشاری با رقم اعشار کمتر %f
برای اعداد اعشاری با رقم اعشار بیشتر %if
برای کاراکتر های تک حرفی %c
برای رشته ای از کاراکتر ها %s

```c
#include <stdio.h>
int main(){
    int a = 1;
    float b = 0.5;
    double c = 0.1;
    char name[] = "Salam";
    char letter = 'A';

    printf("%d", a); // فرمت گذاری اعداد صحیح به صورت %dهست
                     // (a) اول به تابع چاپ گفتیم که یک عدد صحیح چاپ بشه و سپس اون عدد رو مشخص کردیم
    return 0;
}
```

ذخیره و پردازش داده‌های کاربر

شما ابتدا به کمک scanf و fgets داده ها را از کاربر دریافت می کنید آن ها را در حافظه) RAM (سیو می کنید سپس آن ها را پردازش می کنید .

```c
#include <stdio.h>
int main(){
    int a;
    char name[50];
    printf("Adad Bgoo");
    scanf("%d", &a);
    printf("Esm Bgoo");
    scanf("%s", &name);

    return 0;
}
```

فرق fgets با scanf

از آن جایی که در رشته کاراکترهایی که از کاربر می گیریم ممکن هست فضای خالی وجود داشته باشد .)دکمه اسپس یا اینتر(

تابعscanf این فضاهای خالی رو نمیتواند داخل متغیر بریزد ولی fgets این قابلیت رو دارد .

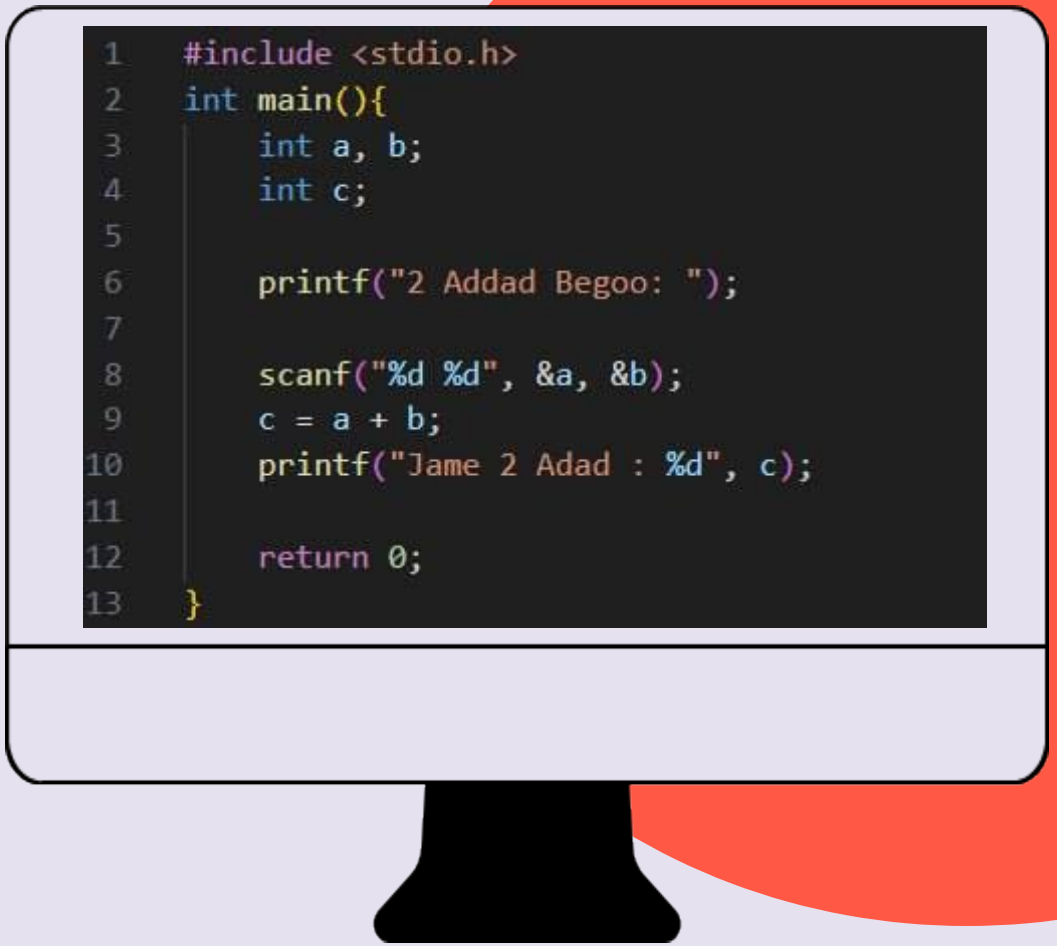
این مشکل وقتی خودش را نشون می دهد که با استفاده از scanf ما از کاربر اول عدد بگیریم و سپس از او یک رشته کاراکتر درخواست کنیم.

فرض میکنیم کاربر عدد 123 رو وارد کرده که به صورت زیر توی متغیر a قرار داده میشه:

['1','2','3','\n']

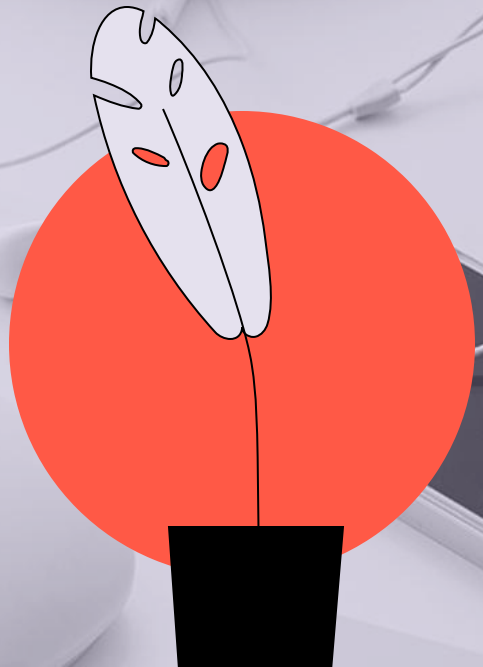
تابع scanfفقط اعداد رو میریزد توی a و نیولاین 'n' در بافر باقی می ماند که باعث می شود تابع scanf بعدی دکمه اینتر را به عنوان ورودی بخواند و به شما اجازه ورود مقداری رو نمی دهد .

راه حل مبتدی برای این مشکل استفاده از تابع getchar است .



```c
#include <stdio.h>
int main(){
    int a, b;
    int c;

    printf("2 Addad Begoo: ");

    scanf("%d %d", &a, &b);
    c = a + b;
    printf("Jame 2 Adad : %d", c);

    return 0;
}
```

بخش پنجم



محاسبات ریاضی و قالب بندی ورودی و خروجی

عملگر ها در ریاضی : این‌ها نمادهایی هستند که برای انجام عملیات پایه‌ی ریاضی در C استفاده می‌شوند .)مثل جمع,تفریق و ...)

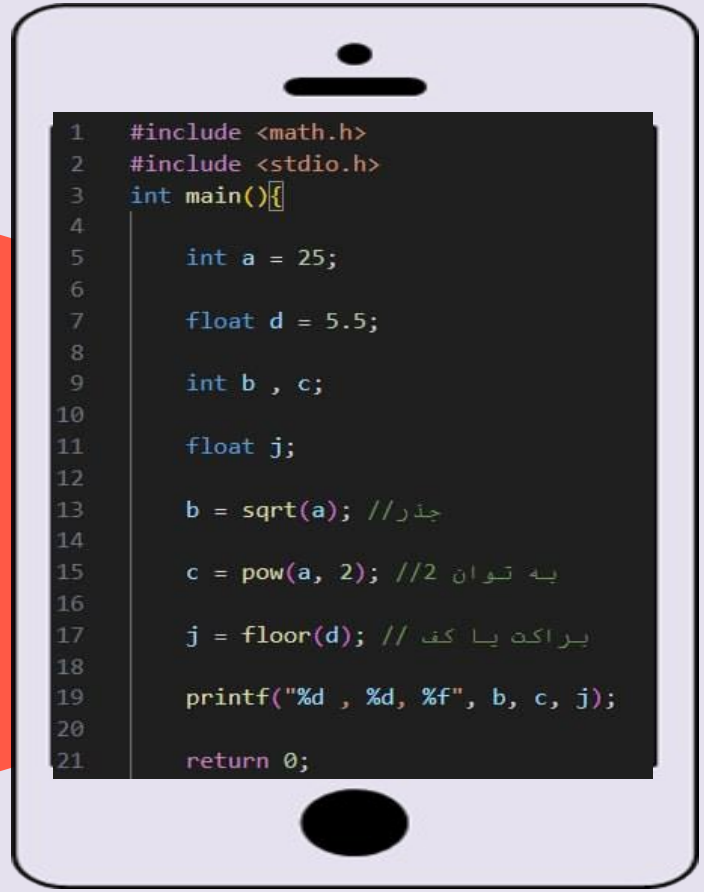
در عمل تقسیم اگر برای ذخیره داده مربوطه اگر به صورت عدد صحیحint تعریف شده باشد, عدد حاصل نیز عدد صحیح است.

باقیمانده یا Modulus
باقیمانده‌ی تقسیم دو عدد صحیح را برمی‌گرداند.

کتابخانه math.h
برای عملیات ریاضی پیچیده‌تر)مانند رادیکال توان توابع مثلثاتی(زبان C یک کتابخانه استاندارد به نام math.h دارد.

نحوه دسترسی به هدرفایل:

```
1    #include <math.h>
```

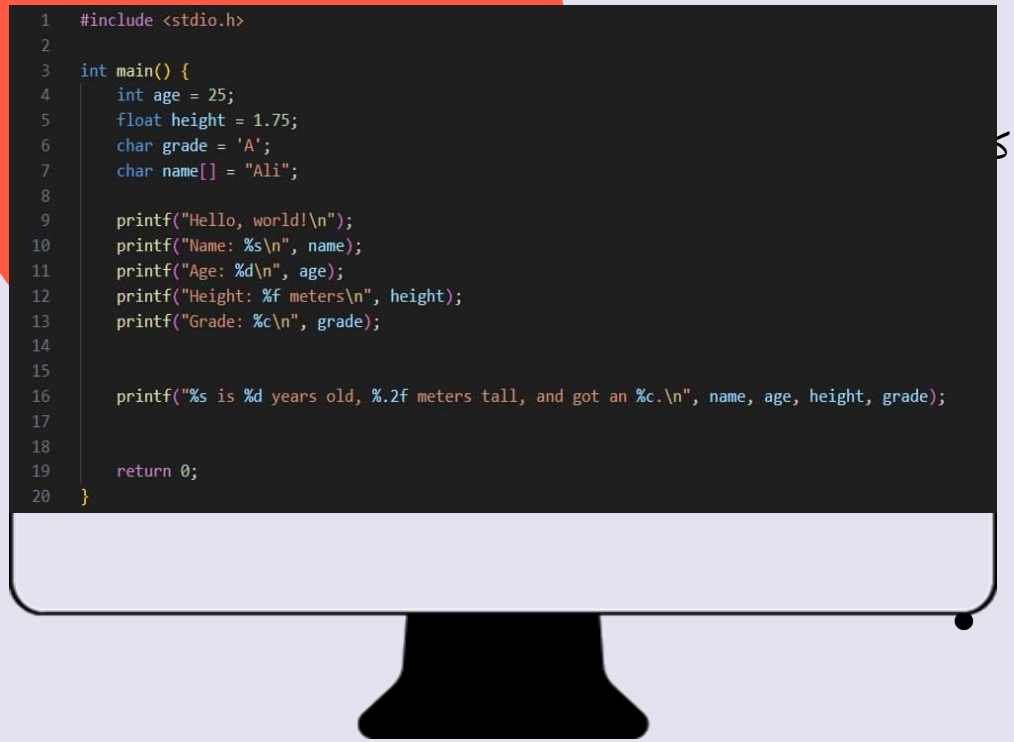


```c
#include <math.h>
#include <stdio.h>
int main(){

    int a = 25;

    float d = 5.5;

    int b , c;

    float j;

    b = sqrt(a); //جذر

    c = pow(a, 2); //2 به توان

    j = floor(d); // براکت یا کف

    printf("%d , %d, %f", b, c, j);

    return 0;
```

توابع رایج کتابخانه math.h

- √ Sqrt : جذر یا ریشه دوم
- √ pow(x, y) : عدد به توان عدد دیگر
- √ floor(x) : براکت یا کف عدد
- √ ceil(x) : سقف یک عدد

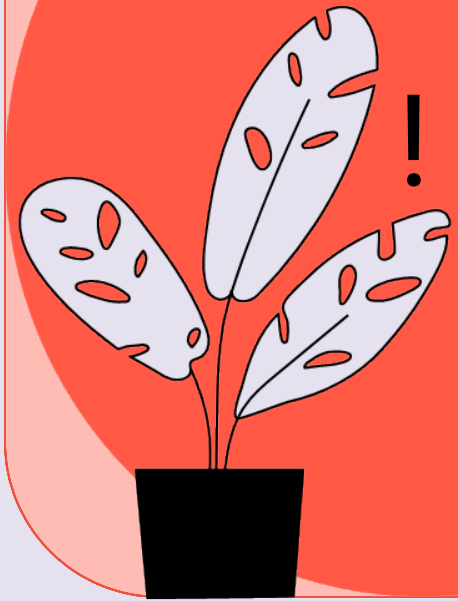
قالب‌بندی خروجی با printf



```c
#include <stdio.h>

int main() {
    int age = 25;
    float height = 1.75;
    char grade = 'A';
    char name[] = "Ali";

    printf("Hello, world!\n");
    printf("Name: %s\n", name);
    printf("Age: %d\n", age);
    printf("Height: %f meters\n", height);
    printf("Grade: %c\n", grade);


    printf("%s is %d years old, %.2f meters tall, and got an %c.\n", name, age, height, grade);


    return 0;
}
```

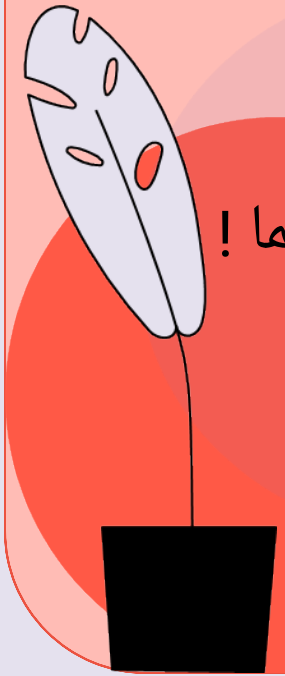
این تابع از مشخص‌کننده‌های فرمت استفاده می‌کند ک
با علامت % شروع می‌شوند تا نشان دهد چه نوع
داده‌ای باید در آنجا چاپ شود.

برای عدد صحیح %d
برای اعداد اعشاری با رقم اعشار کمتر %f
برای اعداد اعشاری با رقم اعشار بیشتر %lf
برای کاراکتر های تک حرفی %c
برای رشته ای از کاراکتر ها %s
برای رفتن به خط جدید بعد از چاپ از \n در انتهای
خروجی استفاده میکنیم .



ممنون از توجه شما !

√
√
√ ممنون از توجه شما !



منابع

W3school
GeeksforGeeks
FreeCodeCamp

√ ممنون از توجه شما !
√
√

منابع

W3school
GeeksforGeeks
FreeCodeCamp

تمرینات :

با استفاده از زبان سیhelloworld را چاپ کنید.
برنامه ای بنویسید که شماره دانشجویی و اسم را بگیرد و هردو را چاپ کند.
برنامه ای بنویسید که اسم و فامیلی را بگیرد و انها را در یک خط چاپ کند.
برنامه ای بنویسید که عرض و طول مستطیلی را بگیرد و مساحتش را محاسبه کند.
برنامه ای بنویسید که سه عدد را بگیرد و میانگین انها را چاپ کند.