C

کورس کامل

۱۷ درس · مبتدی تا پیشرفته

انگلیسی و دری · رایگان

- نحو و نوع‌های دیتا ✓
- حلقه‌ها و تابع‌ها ✓
- آرایه‌ها و رشته‌ها ✓
- اشاره‌گرها و حافظه ✓
- struct · union · لیست پیوندی ✓
- پیش‌پردازنده و هِدرها ✓
- gcc -Wall -g -o ✓
- فایل‌ها و stdlib ✓

همه را یاد بگیرید @MyWebUniversity-Dari

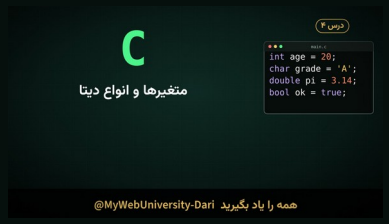
۱۷ درس این کورس



۶. انشعاب: if، else و switch



۵. عملگرها و عبارت‌ها



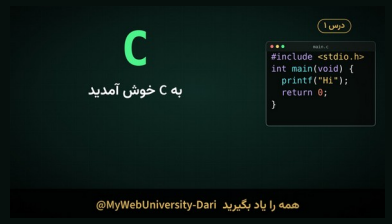
۴. متغیرها و انواع دیتا



۳. ساختار پروگرام، نحو و معنا



۲. کمپایل کردن C با gcc



۱. به برنامه‌نویسی C خوش آمدید



۱۲. حافظهٔ پویا و نشت حافظه



۱۱. اشاره‌گرها



۱۰. رشته‌ها در C



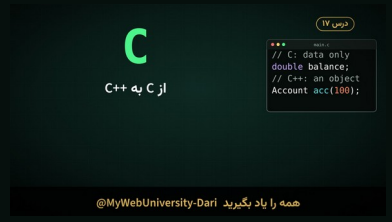
۹. آرایه‌ها و آرایه‌های چندبعدی



۸. تابع‌ها، پروتوتایپ‌ها و حوزه



۷. حلقه‌ها: while، do-while و for



۱۷. طراحی ماژولار، محدودیت‌های C، و راه به سوی C++



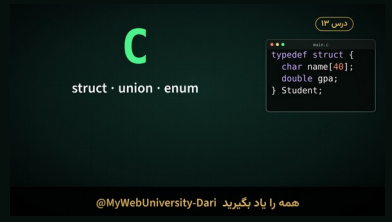
۱۶. کتابخانهٔ استاندارد و فایل‌ها



۱۵. پیش‌پردازنده و فایل‌های هِدر



۱۴. لیست‌های پیوندی



۱۳. struct، union، enum و typedef

یک کورس رایگان از MyWebUniversity

برنامه‌نویسی C

درس ۱

به برنامه‌نویسی C خوش آمدید

چرا C هنوز مهم است، و اولین پروگرام شما



چرا C را یاد بگیریم؟

سریع — نزدیک به سخت‌افزار

همه جا — Linux، Windows، دیتابیس، چیپ

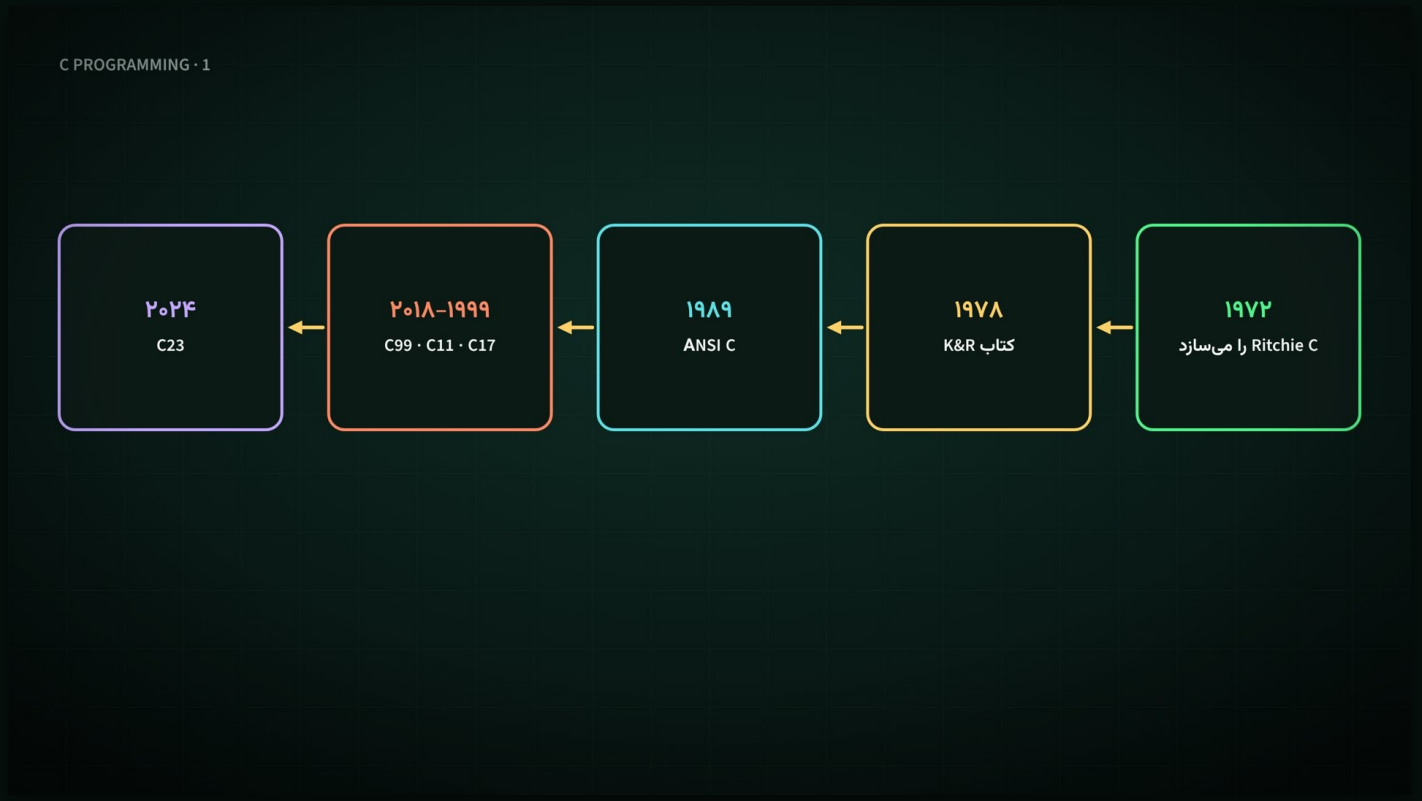
پایه — C++، Java، Go، Python از C گرفتند



۱.۰۱

تاریخچهٔ کوتاه

- ۱۹۷۲ — Ritchie C را می‌سازد
- ۱۹۷۸ — کتاب K&R
- ۱۹۸۹ — ANSI C



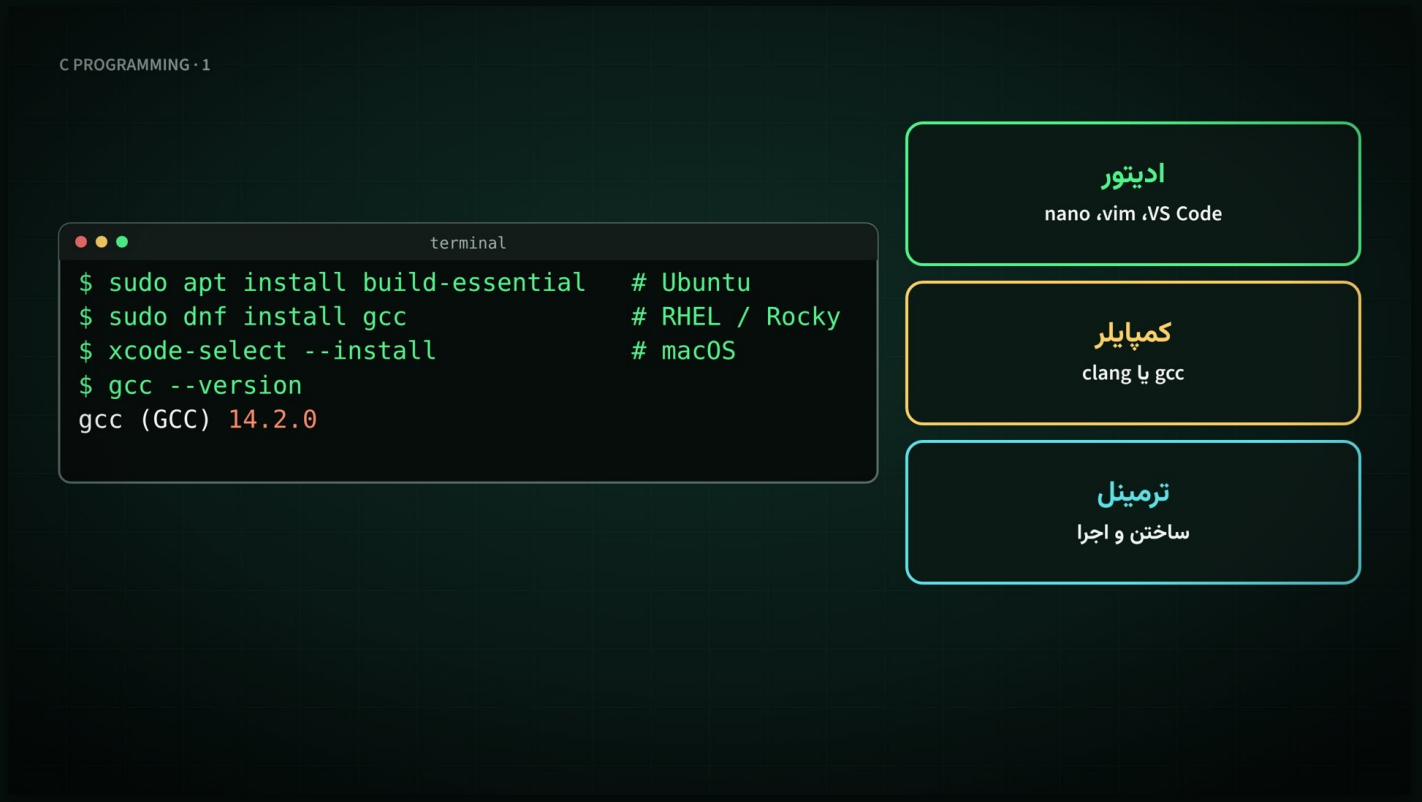
۱.۰۲

به چه چیز ضرورت دارید

ادیتور — VS Code، vim، nano

کمپایلر — gcc یا clang

ترمینل — ساختن و اجرا



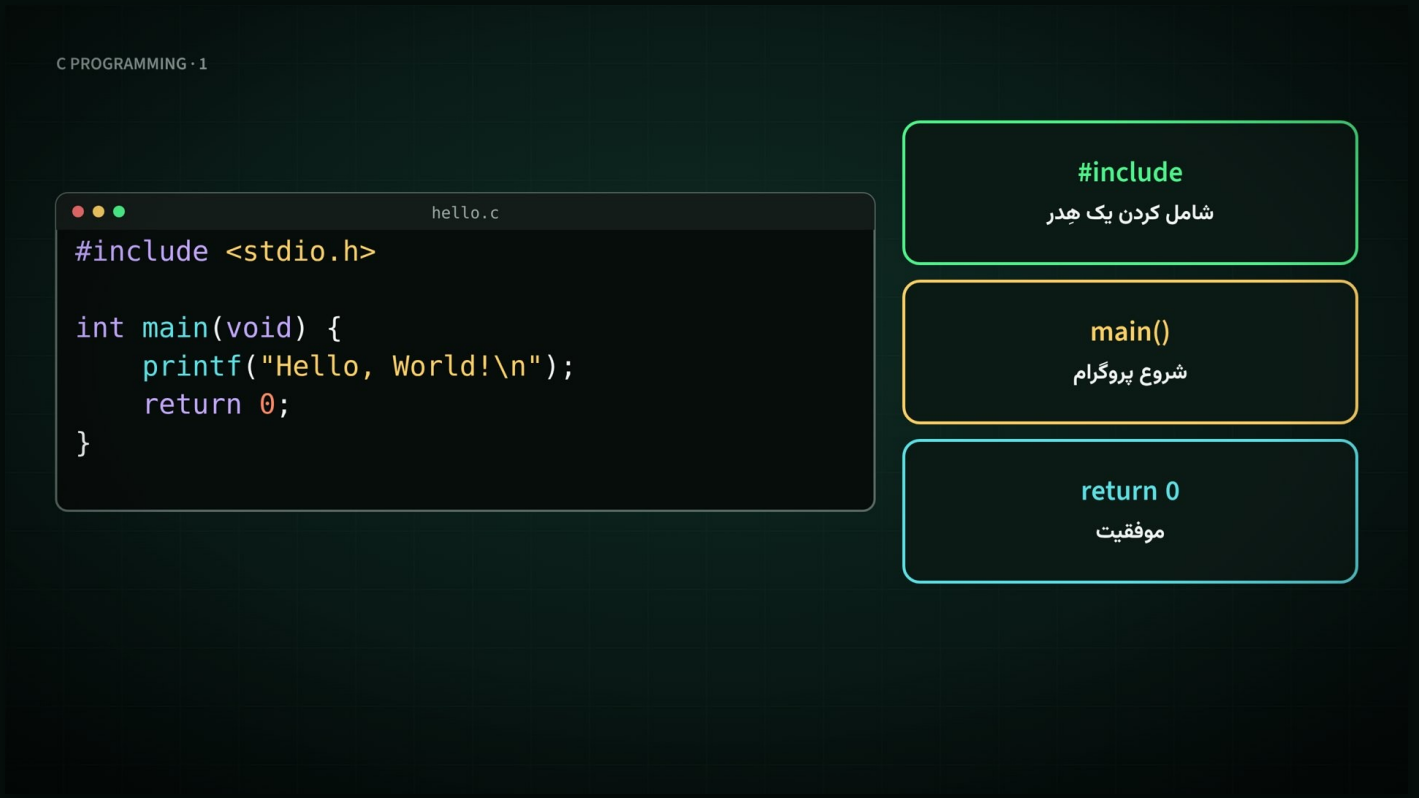
۱.۰۳

اولین پروگرام شما

#include — شامل کردن یک هِدر

main() — شروع پروگرام

return 0 — موفقیت



۱۰۰۴

کمپایل و اجرا

در همان فولدر یک ترمینل باز کنید و بنویسید: gcc hello.c -o hello. gcc کود ...

بعد آن را با ./hello اجرا کنید. جملهٔ Hello, World را روی صفحه می‌بینید.

اگر اشتباه تایپی کنید، کمپایلر نام فایل، شمارهٔ سطر و خطا را نشان می‌دهد. اولین خطا را با ...



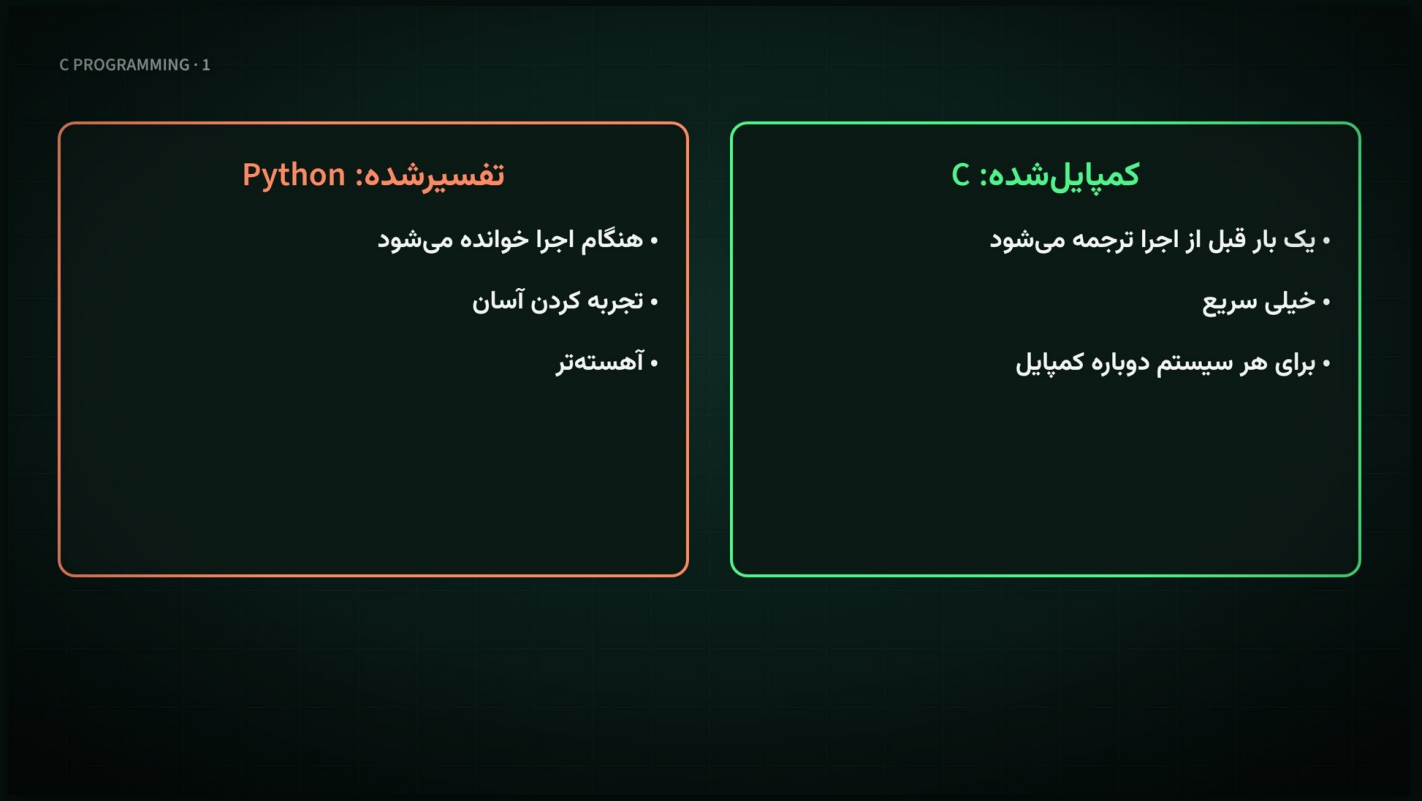
۱.۰۵

کمپایل شده، نه تفسیر شده

C یک زبان کمپایل‌شده است. تمام پروگرام قبل از اجرا به کود ماشین ترجمه می‌شود، به همین دلیل ...

Python معمولاً تفسیر می‌شود: کود شما را هنگام اجرا می‌خواند. تجربه کردن با آن آسان‌تر است، ...

پروگرام کمپایل‌شده تنها روی همان نوع کمپیوتری اجرا می‌شود که برایش ساخته شده. برای اجرا روی ...



۱.۰۶

C امروز کجا اجرا می‌شود

کرنل Linux — Android · سرورها

CPython — خود Python

دیتابیس‌ها — SQLite · PostgreSQL · MySQL



۱.۰۷

نقشهٔ راه شما

۱–۳ — آماده‌سازی · کمپایل · نحو

۴–۸ — انواع · عملگرها · شرط · حلقه · تابع

۹–۱۴ — آرایه · رشته · اشاره‌گر · حافظه · struct · لیست



۱.۰۸

یک کورس رایگان از MyWebUniversity

برنامه‌نویسی C

درس ۲

کمپایل کردن C با gcc

چهار مرحله، فلگ‌های ضروری، و gdb



چهار مرحله

پیش‌پردازش — .i ← #include، #define

کمپایل — C ← اسمبلی .s

اسمبل — ← فایل آبجکت .o



۲.۰۱

نام پروگرام -o:

-o name — انتخاب فایل خروجی

a.out — نام پیش‌فرض



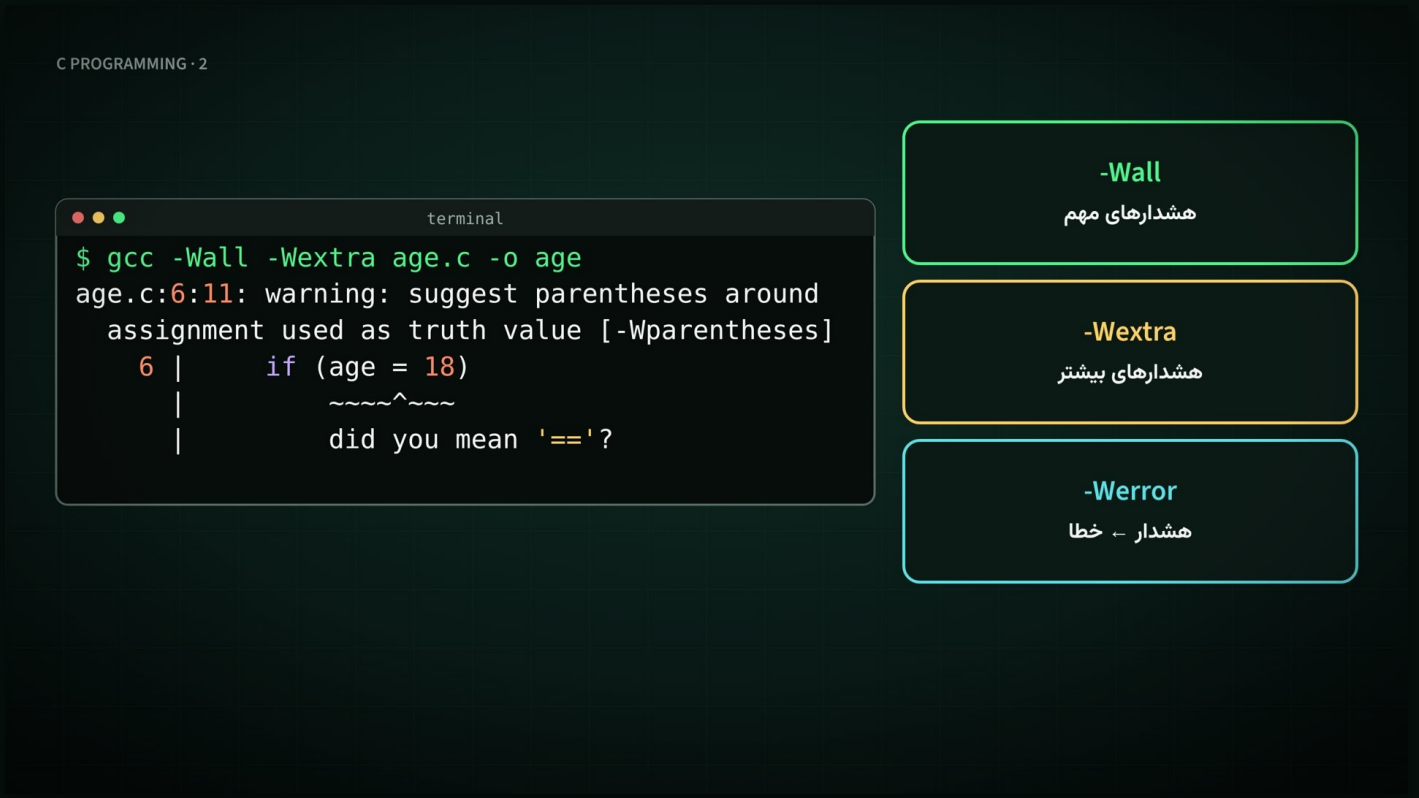
۲.۰۲

-Wall و -Wextra: روشن کردن هشدارها

-Wall — هشدارهای مهم

-Wextra — هشدارهای بیشتر

-Werror — هشدار ← خطا



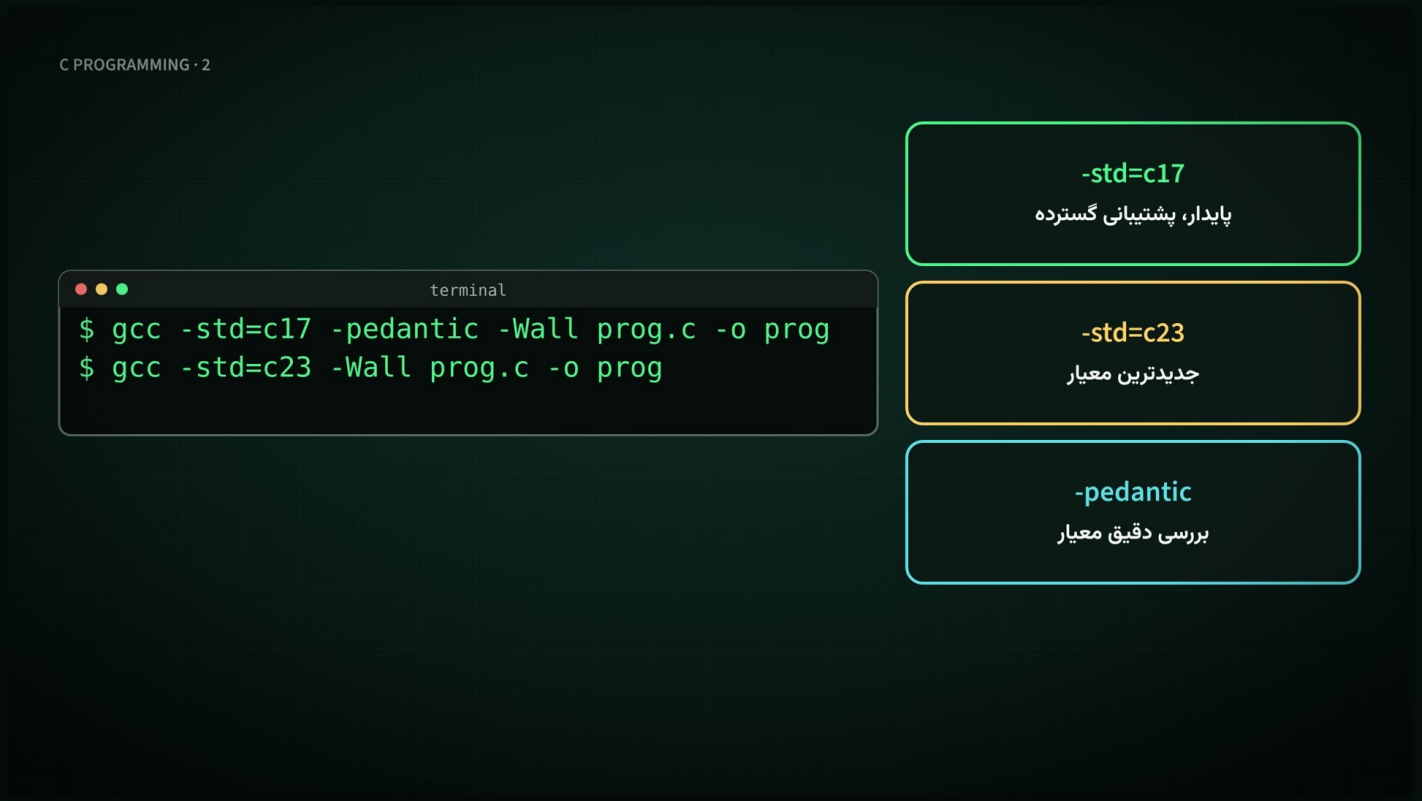
۲.۰۳

انتخاب معیار C با -std:

-std=c17 — پایدار، پشتیبانی گسترده

-std=c23 — جدیدترین معیار

-pedantic — بررسی دقیق معیار



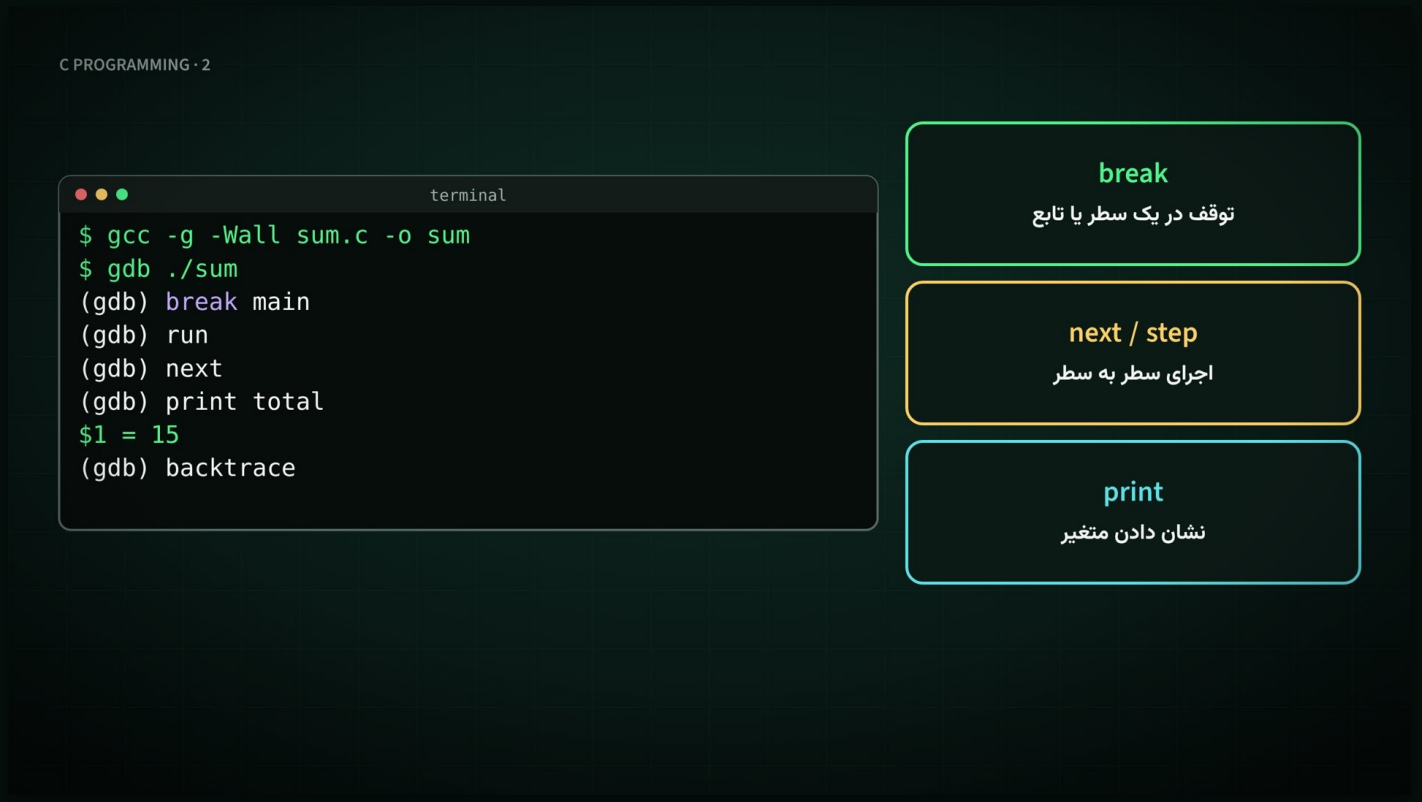
۲۰۰۴

-g و gdb: دیباگ کردن

break — توقف در یک سطر یا تابع

next / step — اجرای سطر به سطر

print — نشان دادن متغیر



۲.۰۵

بهینه‌سازی -O2:

فلگ‌های O- به کمپایلر می‌گویند چقدر بهینه‌سازی کند. O0- یعنی بدون بهینه‌سازی، که ...

O2- پروگرام را خیلی سریع‌تر می‌سازد و انتخاب معمول برای نسخهٔ نهایی است. O3- ...

یک عادت خوب: با g- و O0- دیباگ کنید، و نسخهٔ نهایی را با O2- بسازید.

فلگ	معنی
-O0	بدون بهینه‌سازی — برای دیباگ بهترین
-O2	سریع — نسخهٔ نهایی معمول
-O3	حداکثر بهینه‌سازی سرعت
-Os	بهینه‌سازی برای اندازهٔ کوچک

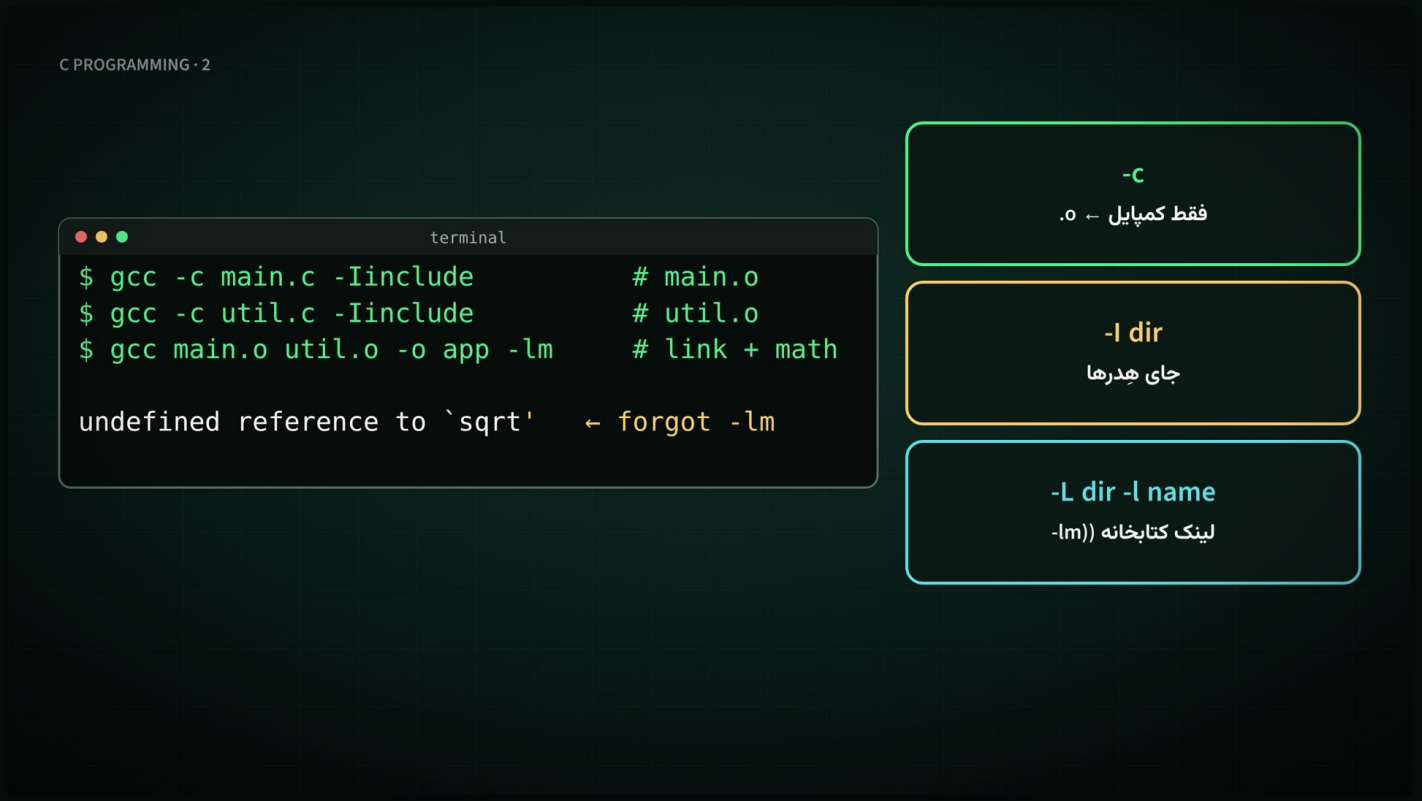
۲.۰۶

-L ،-I ،-c و -I: پروژه‌های بزرگ‌تر

-c — فقط کمپایل ← .o

-I dir — جای هِدرها

-L dir -l name — لینک کتابخانه ((-lm



۲.۰۷

یک فرمان حرفه‌ای

این فرمانی است که برای هر تمرین این کورس استفاده کرده می‌توانید.

این فرمان C17 را انتخاب می‌کند، هشدارهای قوی را روشن می‌کند، معلومات دیباگ را اضافه ...



۲.۰۸

یک کورس رایگان از MyWebUniversity

برنامه‌نویسی C

درس ۳

ساختار پروگرام، نحو و معنا



اجزای سازندهٔ هر پروگرام C

کالبدشناسی یک پروگرام C

هر پروگرام C یک شکل مشابه دارد. در بالا سطرهای پیش‌پردازنده مثل #include هستند. بعد ...

یکی از تابع‌ها باید main نام داشته باشد. سیستم عامل main را صدا می‌کند، و وقتی main ...

داخل یک تابع، دستورها از بالا به پایین، یکی پس از دیگری اجرا می‌شوند.

C PROGRAMMING · 3

area.c

```c
#include <stdio.h>          // 1. preprocessor

#define PI 3.14159          // 2. a constant

double area(double r);      // 3. declaration

int main(void) {           // 4. main function
    double r = 2.0;
    printf("%.2f\n", area(r));
    return 0;
}

double area(double r) {    // 5. definition
    return PI * r * r;
}
```

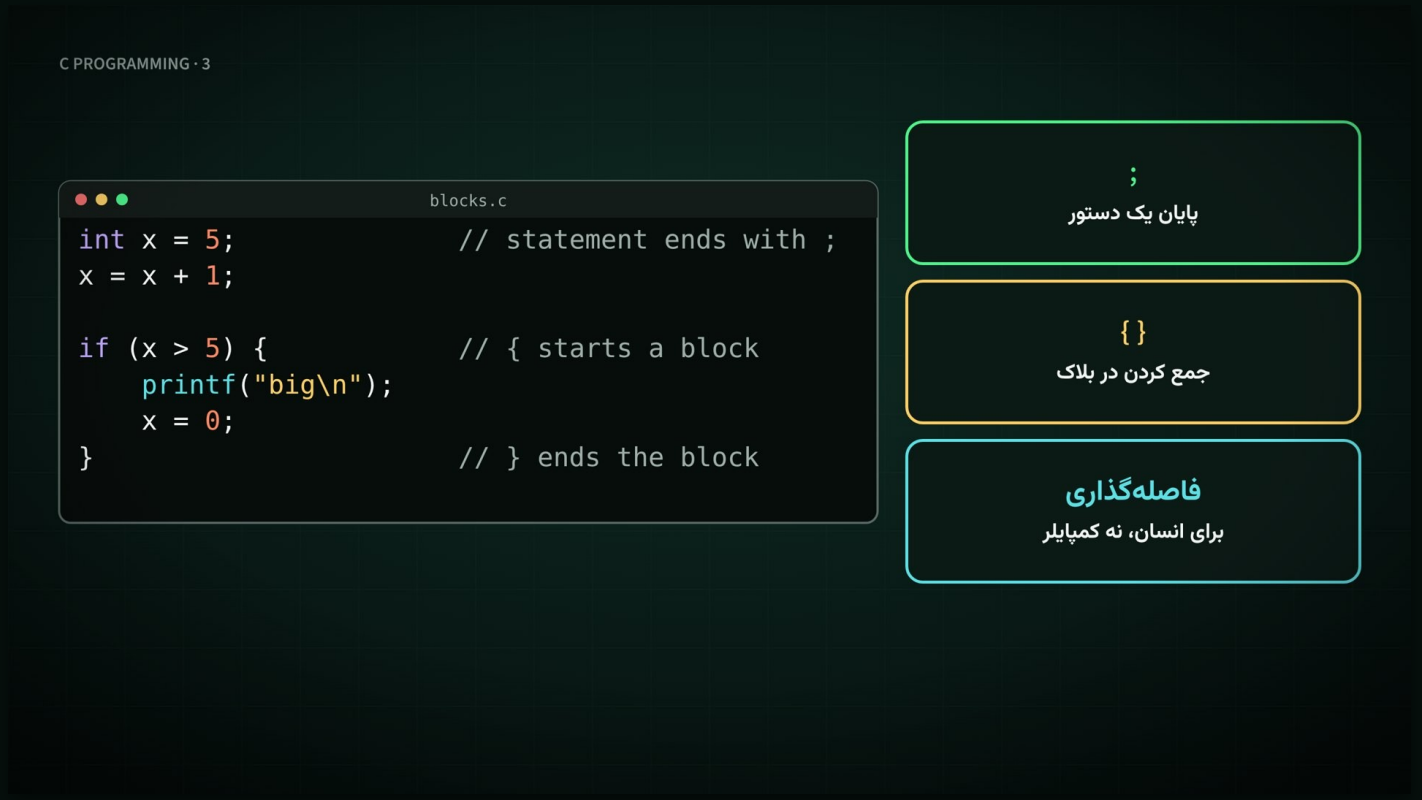
۳.۰۱

دستورها و بلاک‌ها

; — پایان یک دستور

{ } — جمع کردن در بلاک

فاصله‌گذاری — برای انسان، نه کمپایلر



۳.۰۲

توکن‌ها: کلمات C

کمپایلر اول کود شما را به توکن‌ها می‌شکند، یعنی کوچک‌ترین قطعه‌های با معنا، مثل کلمه‌ها در ...

کلمات کلیدی مثل if ،int و return؛ شناسه‌ها، یعنی نام‌هایی که خودتان انتخاب می‌کنید؛ ...

C PROGRAMMING · 3

نوع توکن	مثال‌ها	معنی
کلمهٔ کلیدی	int if while return	رزرو شده توسط C
شناسه	total count area	نام‌های انتخابی شما
ثابت	42 3.14 'A'	مقدارهای ثابت
رشته	"Hello"	متن بین نقل قول
عملگر	+ - * / = == ;	عمل‌ها و علامه‌ها

۳.۰۳

قواعد نام‌گذاری

شناسه‌ها می‌توانند حروف، اعداد و آندرسکور داشته باشند، اما نباید با عدد شروع شوند و ...

C به حروف بزرگ و کوچک حساس است: توتل با حروف کوچک، بزرگ و ترکیبی سه نام متفاوت هستند.

نام‌های با معنا مثل تعداد شاگردان را به جای X انتخاب کنید. خودتان در آینده از این کار ...



۳.۰۴

تبصره‌ها

تبصره‌ها یادداشت‌هایی برای انسان‌ها هستند؛ کمپایلر آن‌ها را نادیده می‌گیرد. دو اسلش ...

اسلش ستاره تبصره‌ای را شروع می‌کند که می‌تواند چند سطر باشد، و ستاره اسلش آن را تمام می‌کند.

تبصره‌های خوب توضیح می‌دهند که کود چرا کاری را می‌کند، نه تنها چه می‌کند.

C PROGRAMMING · 3

comments.c

```c
// one-line comment

/* a comment that
   spans several lines */

int retries = 3;  // why: the network is slow at night
```

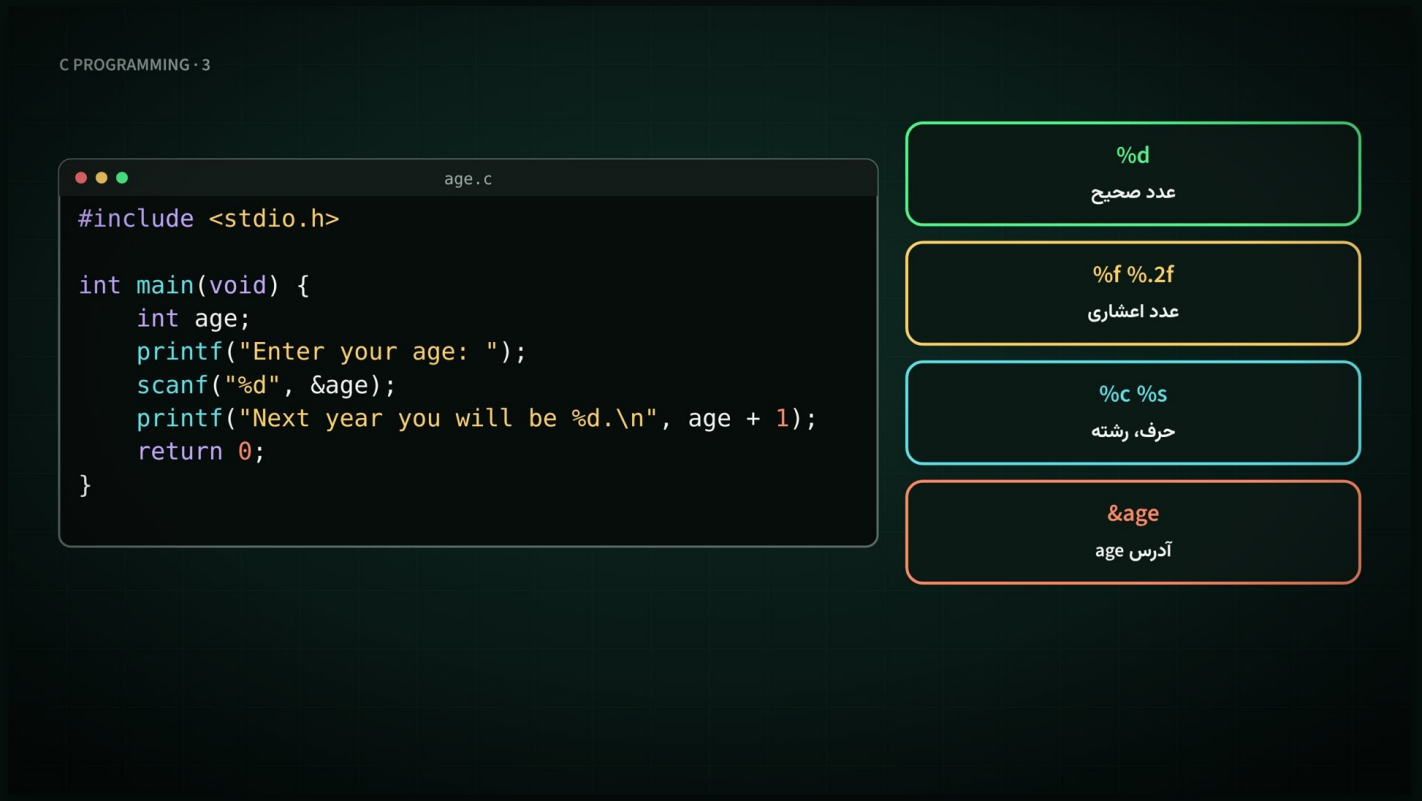
۳.۰۵

scanf و printf

d% — عدد صحیح

f %.2f% — عدد اعشاری

c %s% — حرف، رشته



۳.۰۶

نحو در برابر معنا

نحو دستور زبان است: جای سیمی‌کالن، قوس‌ها و پرانتزها. اگر نحو اشتباه باشد، کمپایلر از ...

معنا یعنی مفهوم: کود هنگام اجرا واقعاً چه می‌کند. کود می‌تواند نحو کاملاً درست داشته باشد و ...

اینجا هر دو سطر کمپایل می‌شوند، اما تقسیم دو عدد صحیح بخش اعشاری را حذف می‌کند، پس اوسط ...



۳.۰۷

کود خوانا بنویسید

فاصلهٔ چهارتایی — ساختار را نشان دهید

هر سطر یک ایده — خواندن و دیباگ آسان

نام‌های روشن — student_count، نه x



۳.۰۸

یک کورس رایگان از MyWebUniversity

برنامه‌نویسی C

درس ۴

متغیرها و انواع دیتا

int، char، float، double، sizeof، const و تبدیل نوع



متغیر یک قطی نام‌دار است

متغیر یک جای نام‌دار در حافظه است که یک مقدار را نگه می‌دارد. در C باید قبل از استفاده نوع ...

نوع به کمپایلر می‌گوید چند بایت رزرو کند و آن‌ها را چطور تفسیر کند. اینجا اِیج یک int است ...

C PROGRAMMING · 4

main.c

```c
int    age   = 20;
char   grade = 'A';
double price = 9.99;
```



۴.۰۱

نوع‌های اصلی

C چند نوع اصلی دارد. char یک حرف یا یک عدد کوچک را نگه می‌دارد. int اعداد صحیح را نگه ...

تغییردهنده‌ها محدوده را تغییر می‌دهند: short و long اندازه را تغییر می‌دهند، و unsigned ...

نوع	اندازهٔ معمول	نگه می‌دارد
char	1 byte	یک حرف، از ۱۲۸- تا ۱۲۷
int	4 bytes	اعداد صحیح، حدود ۲٫۱ میلیارد ±
long long	8 bytes	اعداد صحیح خیلی بزرگ
float	4 bytes	اعشاری، حدود ۷ رقم
double	8 bytes	اعشاری، حدود ۱۵ رقم
unsigned int	4 bytes	صفر تا حدود ۴٫۲ میلیارد

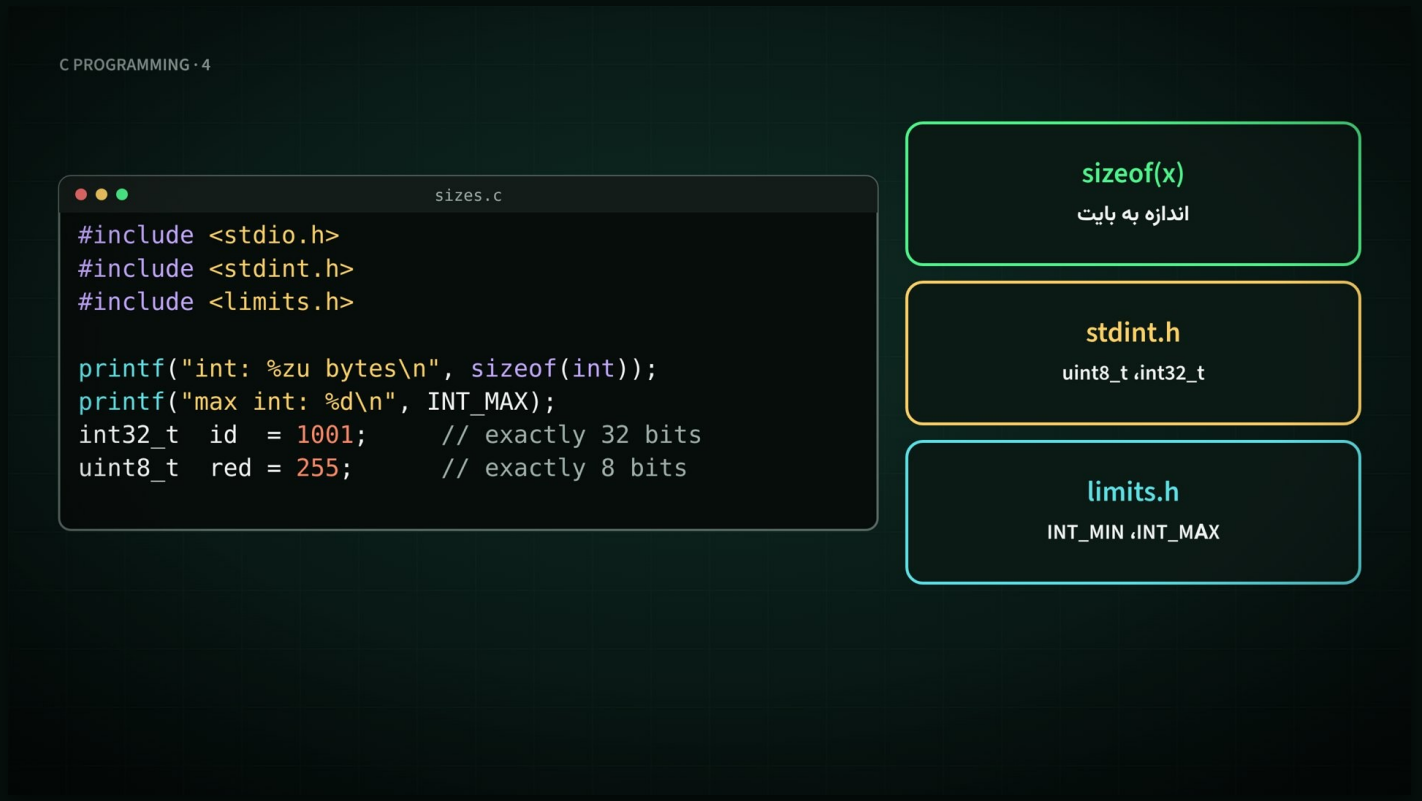
۴.۰۲

sizeof و اندازه‌های دقیق

sizeof(x) — اندازه به بایت

stdint.h — int32_t، uint8_t

limits.h — INT_MAX، INT_MIN

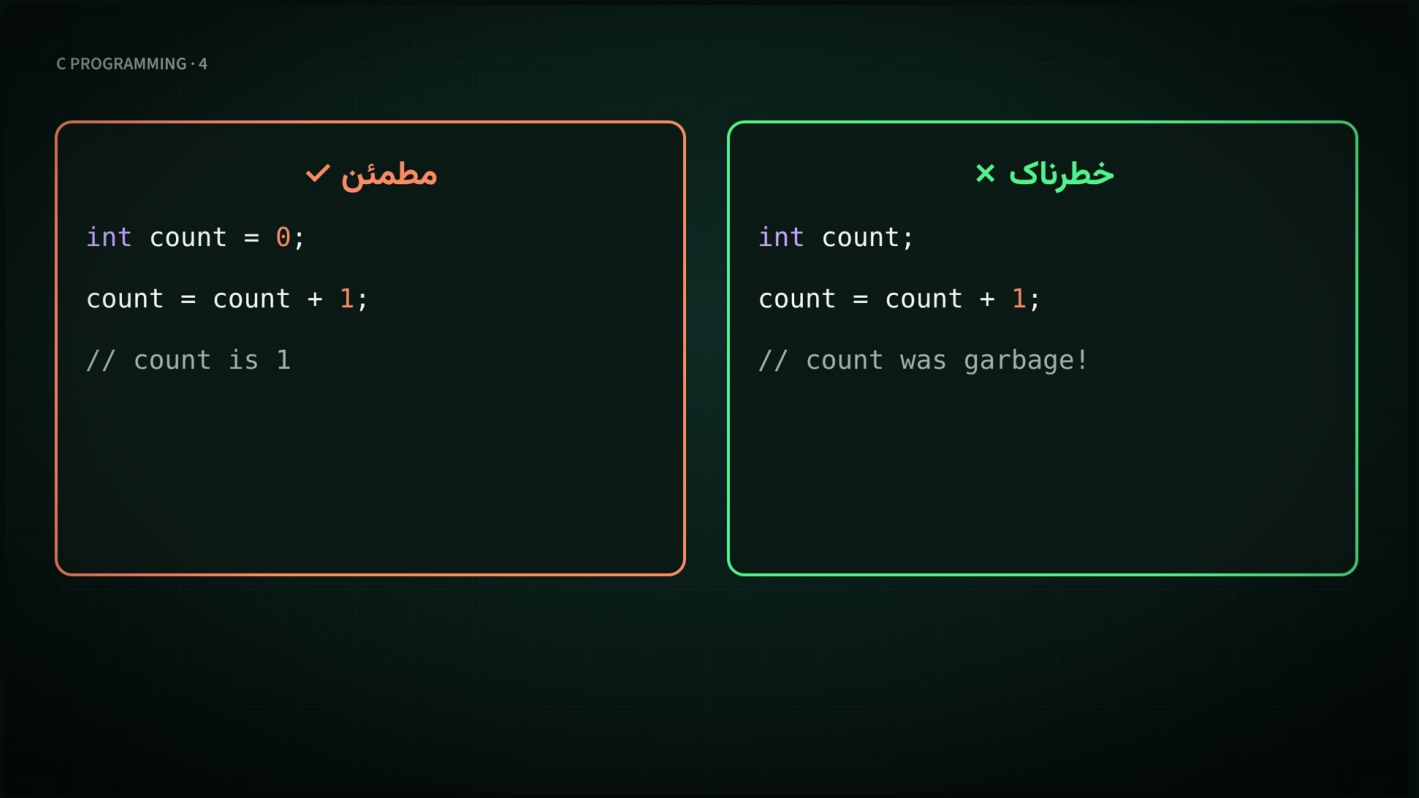


۴۰۰۳

اعلان و مقداردهی

اعلان یک متغیر حافظه رزرو می‌کند. مقداردهی به آن مقدار اول می‌دهد. در C، متغیر محلی که ...

خواندن یک متغیر بی‌مقدار یک باگ کلاسیک است، پس همیشه به متغیرها مقدار شروع بدهید.

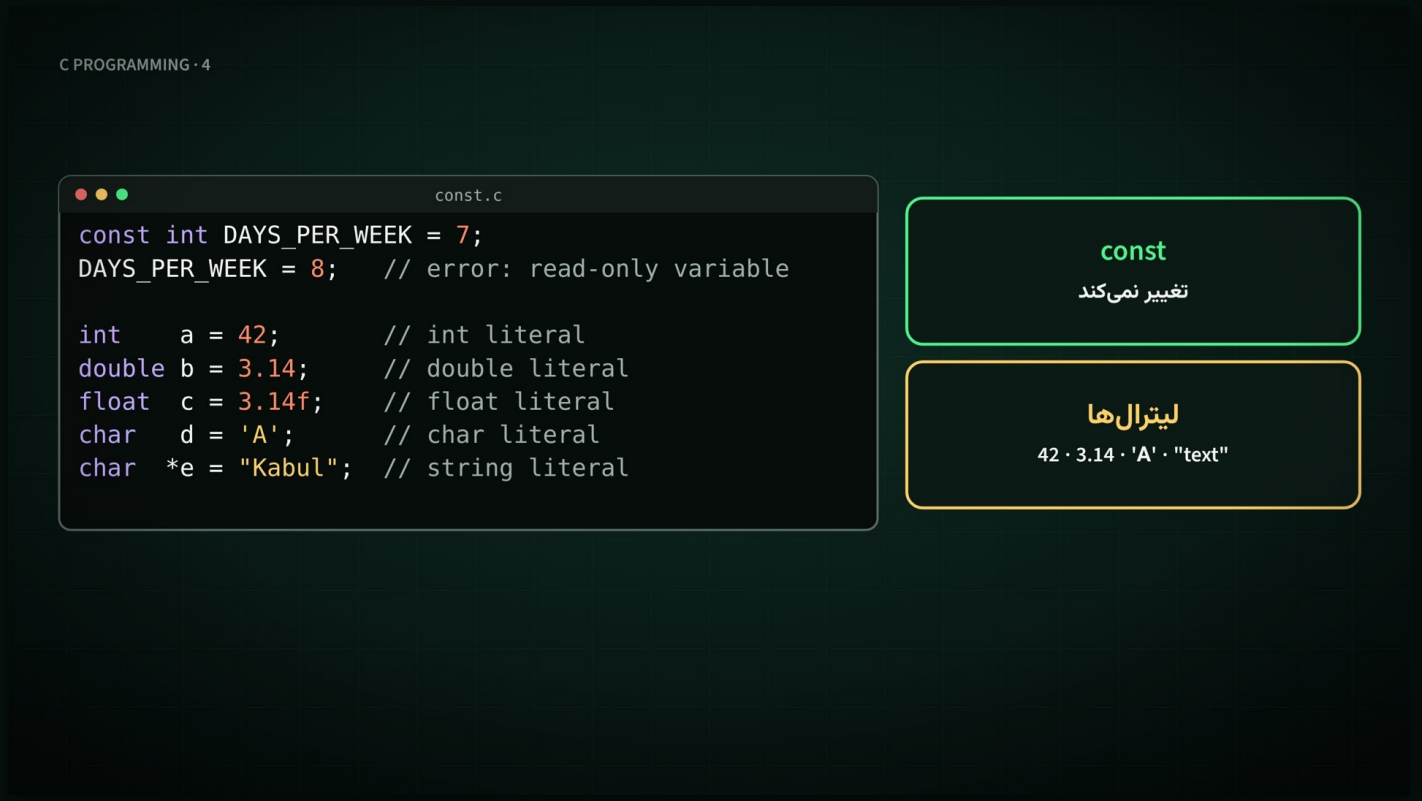


۴۰۴

ثابت‌ها و لیترال‌ها

const — تغییر نمی‌کند

لیترال‌ها — 42 · 3.14 · 'A' · "text"



۴.۰۵

حروف هم عدد هستند

در C، char در واقع یک عدد کوچک است. حرف 'A' در جدول ASCII به شکل شصت و پنج ذخیره می‌شود، بی ...

به همین دلیل می‌توانید با حروف حساب کنید: 'A' + 1 می‌شود 'B'. متن به زبان‌های دیگر، مثل ...

C PROGRAMMING · 4

main.c

```c
char c = 'A';
printf("%c = %d\n", c, c);      // A = 65
printf("%c\n", c + 1);          // B
```



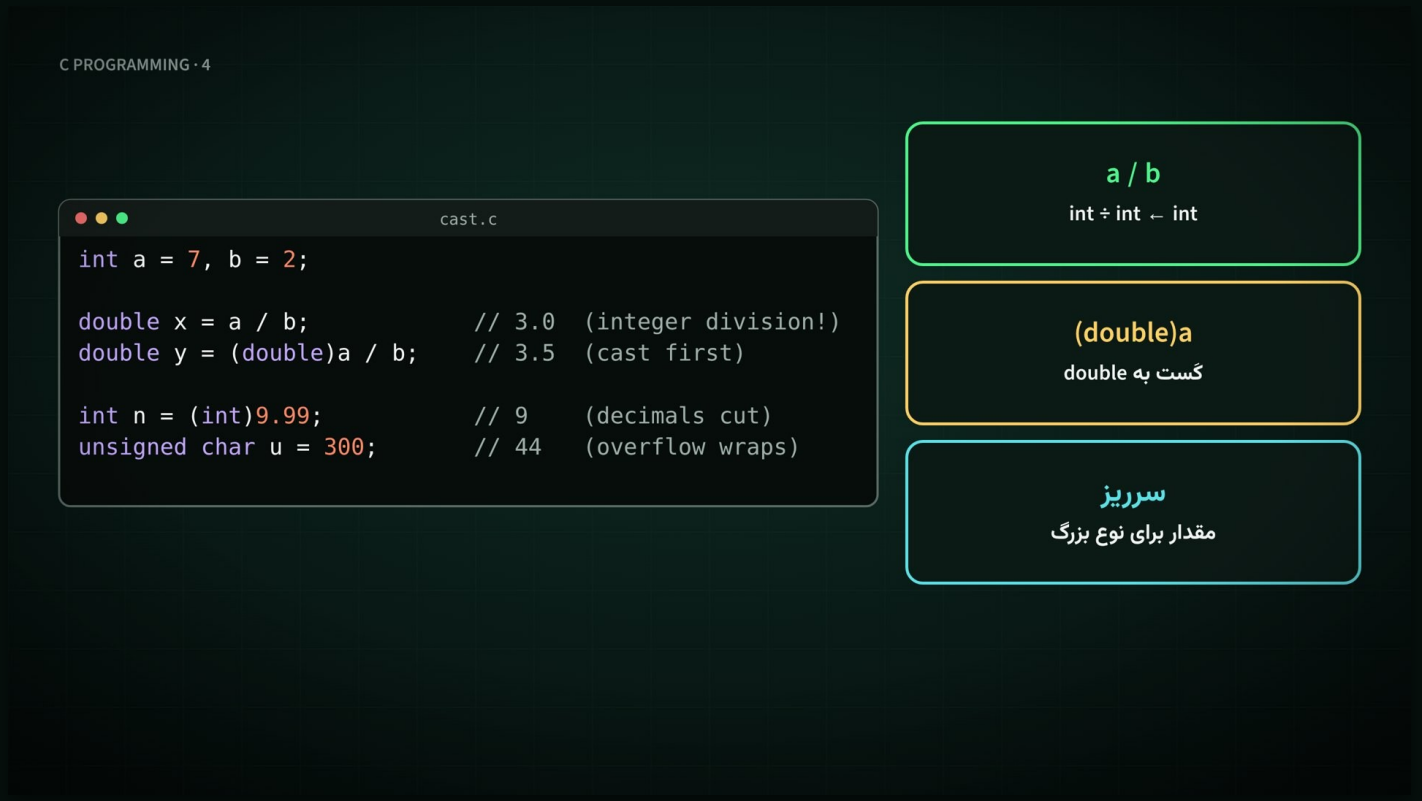
۴.۰۶

تبدیل نوع و کَست

a / b — int ÷ int ← int

(double)a — کَست به double

سرریز — مقدار برای نوع بزرگ



۴.۰۷

درست و نادرست

C کلاسیک نوع Boolean ندارد: صفر یعنی نادرست، و هر مقدار دیگر یعنی درست.

از C99 به بعد می‌توانید فایل اِستاندارد bool را شامل کنید و bool، true و false بنویسید که کود ...

C PROGRAMMING · 4

bool.c

```c
#include <stdbool.h>

bool is_adult = age >= 18;
if (is_adult) {
    printf("Welcome!\n");
}

if (5)   { /* runs: 5 is true  */ }
if (0)   { /* never runs       */ }
```

۴.۰۸

یک کورس رایگان از MyWebUniversity

برنامه‌نویسی C

درس ۵

عملگرها و عبارت‌ها

حسابی، مقایسه، منطقی، بیتی و اولویت



عملگرهای حسابی

C PROGRAMMING · 5

عملگر	مثال	نتیجه
+	17 + 5	22
-	17 - 5	12
*	17 * 5	85
/	17 / 5	۳ (تقسیم صحیح)
%	17 % 5	۲ (باقی‌مانده)

C عملگرهای حسابی آشنا را دارد: جمع، تفریق، ستاره برای ضرب و اسلش برای تقسیم.

علامت درصد عملگر باقی‌مانده یا ماجولو است. هفده درصد پنج می‌شود دو. برای بررسی اعداد جفت یا ...

به یاد داشته باشید: تقسیم دو عدد صحیح عدد صحیح می‌دهد. هفده تقسیم بر پنج سه است، نه سه ...

۵.۰۱

مقداردهی و میان‌بُرها

= — ذخیرهٔ مقدار

+= -= *= /= — به‌روزرسانی

++ -- — یکی اضافه یا کم

C PROGRAMMING · 5

assign.c

```c
int x = 10;
x += 5;      // x = x + 5   → 15
x *= 2;      // x = x * 2   → 30
x %= 7;      // x = x % 7   → 2

int i = 5;
int a = i++; // a = 5, then i = 6
int b = ++i; // i = 7, then b = 7
```

=
ذخیرهٔ مقدار

+= -= *= /=
به‌روزرسانی

++ --
یکی اضافه یا کم

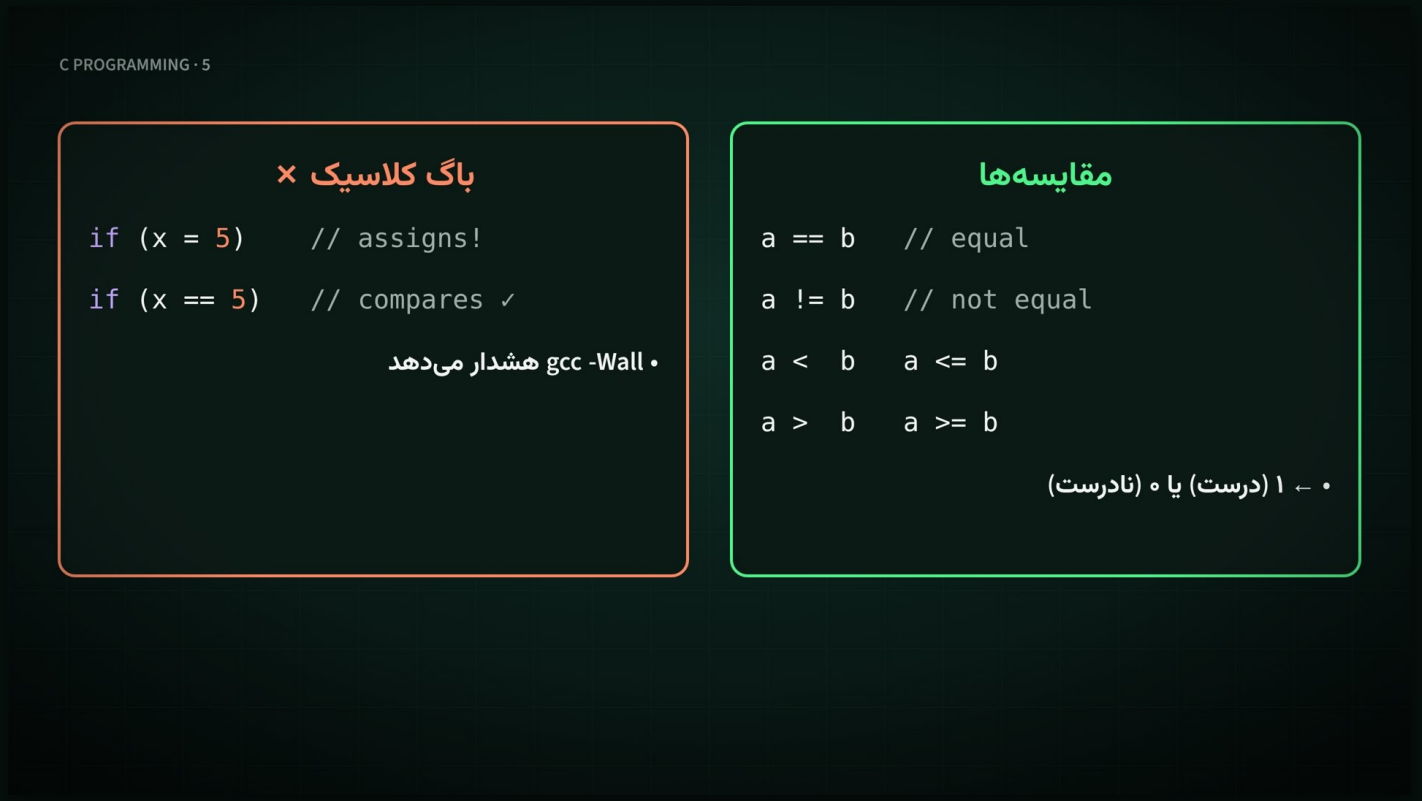
۵.۰۲

عملگرهای مقایسه

مقایسه‌ها یک را برای درست و صفر را برای نادرست می‌دهند. دو علامت مساوی برابری را بررسی ...

کوچک‌تر، بزرگ‌تر، و نسخه‌های یا مساوی آن‌ها هم وجود دارند.

مشهورترین باگ C نوشتن یک علامت مساوی به جای دو علامت داخل if است. این مقدار می‌دهد به جای ...



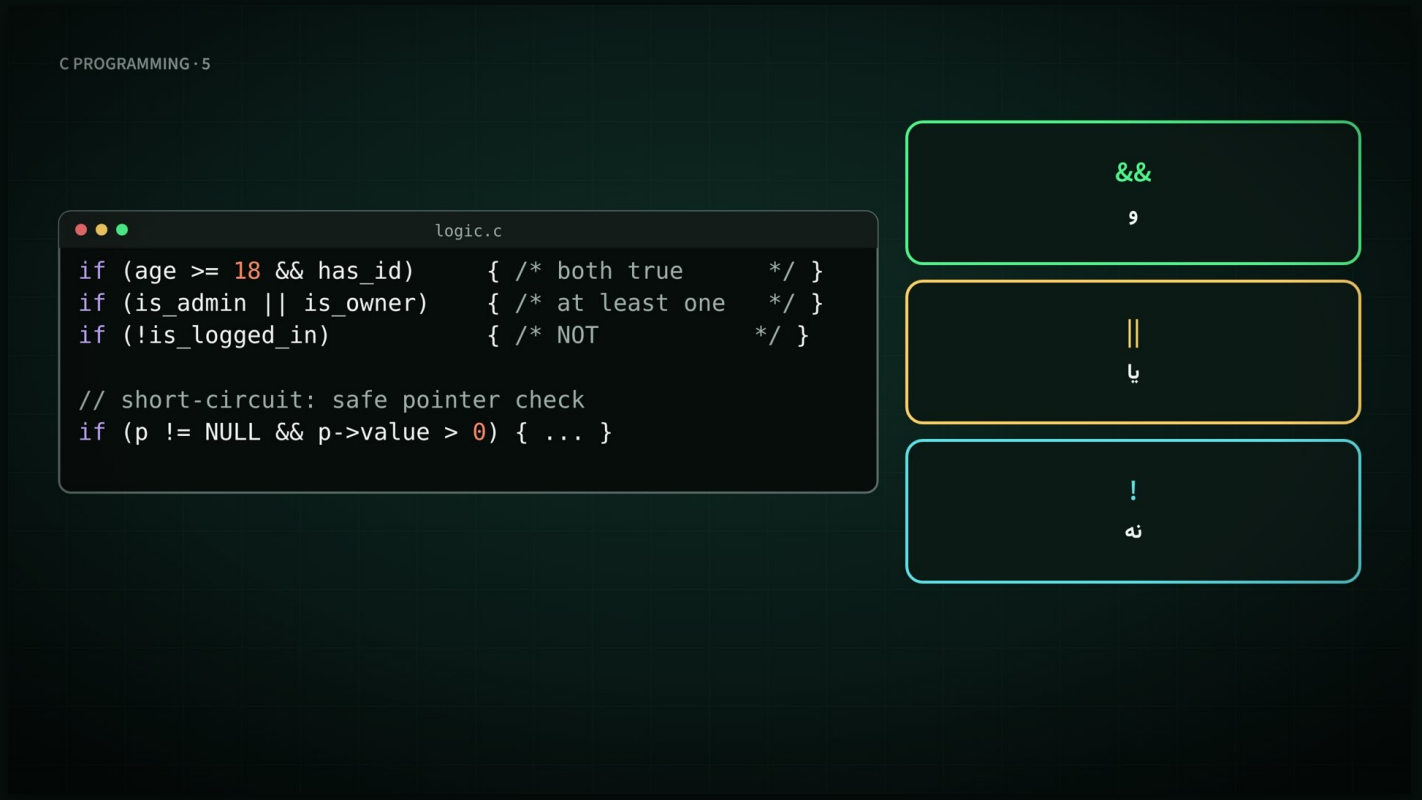
۵.۰۳

عملگرهای منطقی

&& — و

|| — یا

! — نه



۵.۰۴

عملگرهای بیتی

عملگرهای بیتی روی بیت‌های جداگانهٔ یک عدد کار می‌کنند. یک اَمپرسَند «و» بیتی است، یک خط ...

شیفت به چپ با توان‌های دو ضرب می‌کند و شیفت به‌راست تقسیم. این عملگرها برای فلگ‌ها، ...

C PROGRAMMING · 5

عملگر	مثال	باینری	نتیجه
&	12 & 10	1100 & 1010	8
|	12 | 10	1100 | 1010	14
^	12 ^ 10	1100 ^ 1010	6
<<	3 << 2	0011 → 1100	12
>>	12 >> 2	1100 → 0011	3

۵.۰۵

اولویت و پرانتز

وقتی یک عبارت چند عملگر دارد، اولویت تعیین می‌کند کدام اول اجرا شود. ضرب و تقسیم مثل ریاضی ...

مقایسه‌ها قبل از «و» می‌آیند، و «و» قبل از «یا». مقداردهی آخر از همه است.

لازم نیست تمام جدول را حفظ کنید. وقتی شک دارید، از پرانتز استفاده کنید. پرانتز قصد شما را ...

C PROGRAMMING · 5

precedence.c

```c
int r1 = 2 + 3 * 4;         // 14, not 20
int r2 = (2 + 3) * 4;       // 20

// highest → lowest (simplified):
//  ()  ++ --  !   * / %   + -   << >>
//  < <= > >=   == !=   &&   ||   = +=
```

۵.۰۶

عملگر شرطی

عملگر شرطی، که عملگر سه‌تایی هم گفته می‌شود، یک if و else کوتاه است که یک مقدار می‌دهد. ...

برای انتخاب‌های کوچک مثل گرفتن بزرگ‌ترِ دو عدد عالی است، اما برای چیزهای طولانی‌تر، یک ...

C PROGRAMMING · 5

ternary.c

```c
int max = (a > b) ? a : b;

printf("%s\n", (n % 2 == 0) ? "even" : "odd");
```

شرط ? مقدار_درست : مقدار_نادرست

۵.۰۷

یک کورس رایگان از MyWebUniversity

برنامه‌نویسی C

درس ۶

انشعاب: if، else و switch

به پروگرام تصمیم‌گیری یاد بدهید

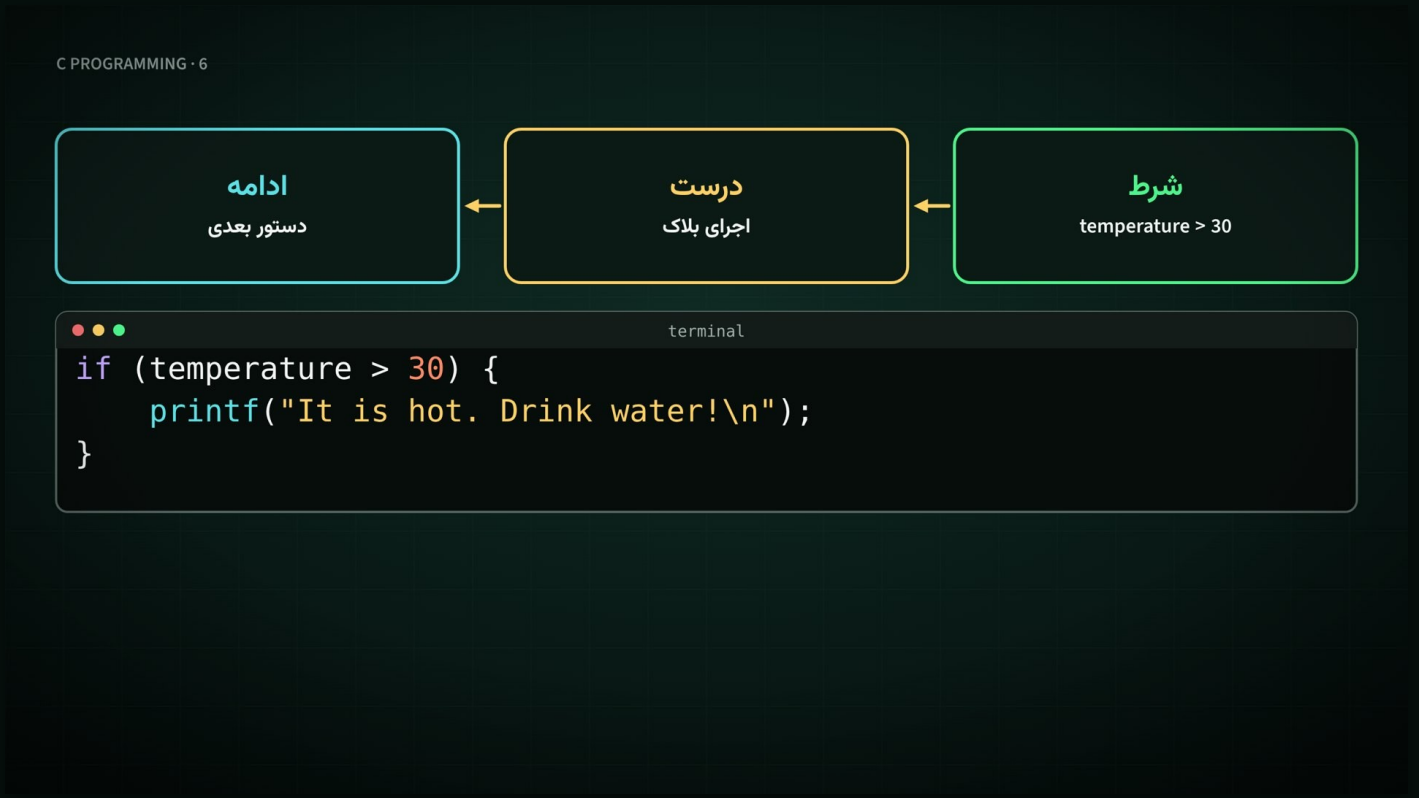


دستور if

شرط — temperature > 30

درست — اجرای بلاک

ادامه — دستور بعدی



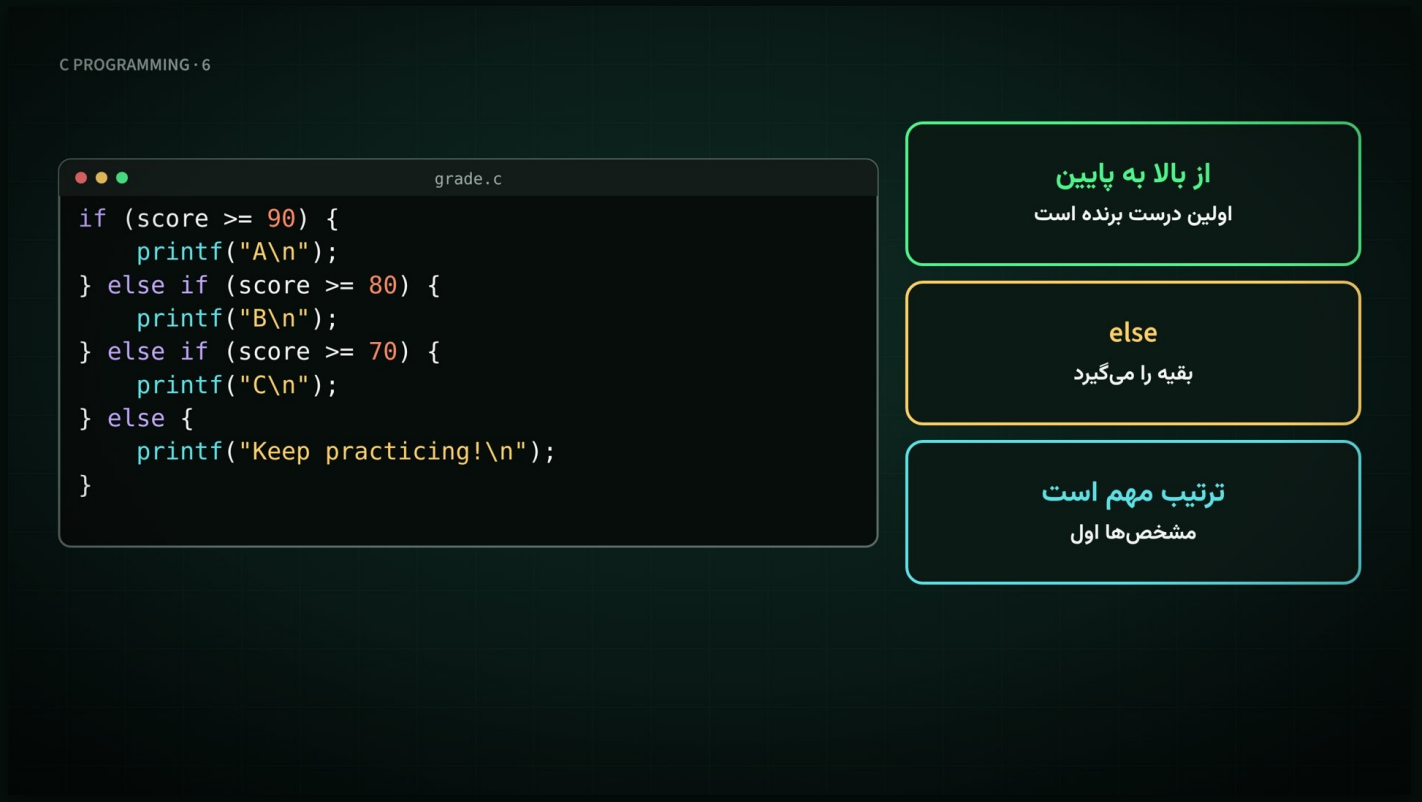
۶.۰۱

if، else if، else

از بالا به پایین — اولین درست برنده است

else — بقیه را می‌گیرد

ترتیب مهم است — مشخص‌ها اول

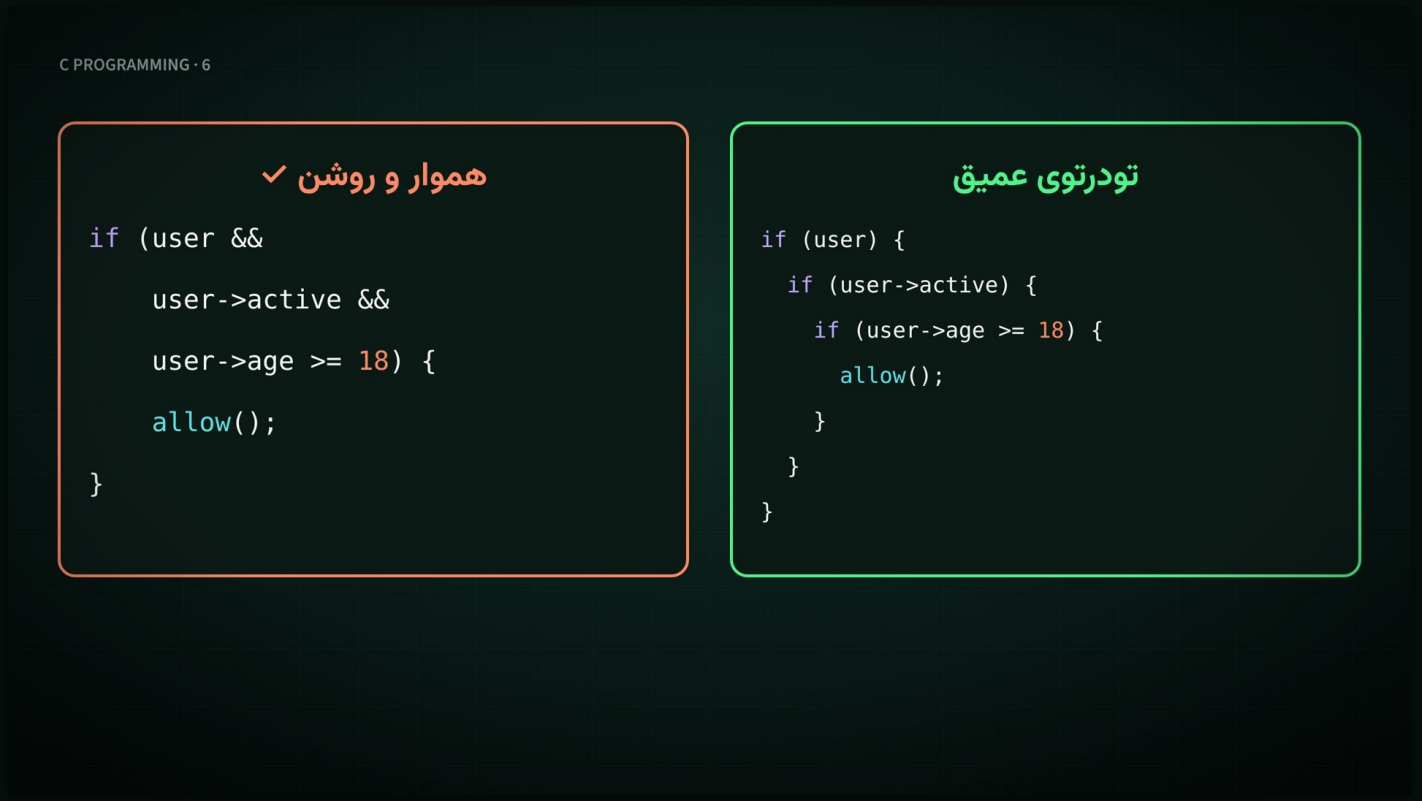


۶.۰۲

شرط‌های تودرتو

می‌توانید یک if را داخل if دیگر بگذارید. این را تودرتو کردن می‌گویند.

اما تودرتوی عمیق زود خواندنش مشکل می‌شود. اکثراً می‌توانید شرط‌ها را با «و» ترکیب کنید، یا ...



۶.۰۳

دستور switch

case — یک مقدار ثابت

break — خروج از switch

default — هیچ جوری

C PROGRAMMING · 6

menu.c

```c
switch (choice) {
    case 1:
        printf("New file\n");
        break;
    case 2:
        printf("Open file\n");
        break;
    case 'q':
        printf("Quit\n");
        break;
    default:
        printf("Unknown option\n");
}
```

case
یک مقدار ثابت

break
خروج از switch

default
هیچ جوری

۶.۰۴

افتادن به case بعدی

اگر break را فراموش کنید، اجرا به case بعدی می‌افتد. گاهی این یک باگ است و گاهی مفید، مثلاً ...

وقتی عمداً اجازه می‌دهید بیفتد، یک تبصره بنویسید تا خوانندهٔ بعدی بداند که عمدی بوده.

C PROGRAMMING · 6

days.c

```c
switch (day) {
    case 6:
    case 7:                 // grouped on purpose
        printf("Weekend\n");
        break;
    case 1: case 2: case 3: case 4: case 5:
        printf("Workday\n");
        break;
}
```

۶.۰۵

اشتباهات رایج

مواظب این اشتباهات کلاسیک باشید: استفاده از یک علامت مساوی به جای دو؛ گذاشتن سیمی‌کالن درست ...

اعداد اعشاری تقریبی ذخیره می‌شوند، پس صفر اعشاریه یک جمع صفر اعشاریه دو دقیقاً صفر اعشاریه ...

C PROGRAMMING · 6

pitfalls.c

```c
if (x = 0) { }            // ✗ assigns 0, never runs
if (x > 5); {             // ✗ stray ';' ends the if
    run();                 //   this always runs!
}
if (0.1 + 0.2 == 0.3) { } // ✗ false!
if (fabs(a - b) < 1e-9) { } // ✓ compare with tolerance
```

۶.۰۶

مثال: سال کبیسه

بیایید همه را یکجا کنیم. یک سال کبیسه است اگر بر چهار قابل تقسیم باشد، مگر سال‌هایی که بر ...

این یک شرط از ماجولو، مقایسه، «و»، «یا» و پرانتز استفاده می‌کند. آن را با دو هزار و بیست و ...

C PROGRAMMING · 6

leap.c

```c
#include <stdio.h>

int main(void) {
    int year;
    printf("Year: ");
    scanf("%d", &year);
    if ((year % 4 == 0 && year % 100 != 0) || year % 400 == 0) {
        printf("%d is a leap year\n", year);
    } else {
        printf("%d is not a leap year\n", year);
    }
    return 0;
}
```

۶.۰۷

یک کورس رایگان از MyWebUniversity

برنامه‌نویسی C

درس ۷

حلقه‌ها: while، do-while و for

تکرار کار بدون تکرار کود



حلقهٔ while

اول شرط — شاید هیچ بار اجرا نشود

چیزی را تغییر دهید — وگرنه هرگز متوقف نمی‌شود

C PROGRAMMING · 7

countdown.c

```c
int n = 5;
while (n > 0) {          // checked before each round
    printf("%d ", n);
    n--;                 // moves toward the end
}
printf("Liftoff!\n");
// 5 4 3 2 1 Liftoff!
```

اول شرط
شاید هیچ بار اجرا نشود

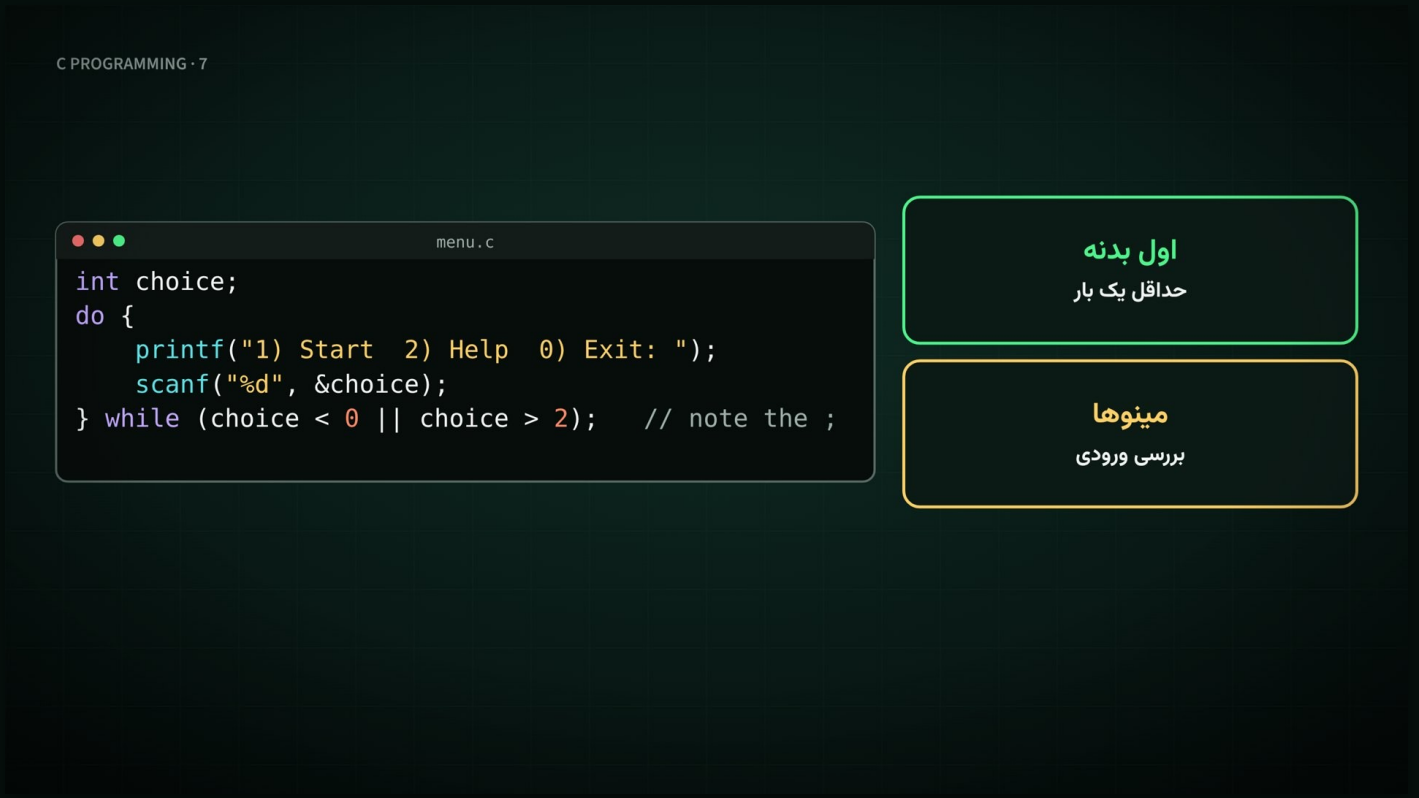
چیزی را تغییر دهید
وگرنه هرگز متوقف نمی‌شود

۷.۰۱

حلقهٔ do-while

اول بدنه — حداقل یک بار

مینوها — بررسی ورودی



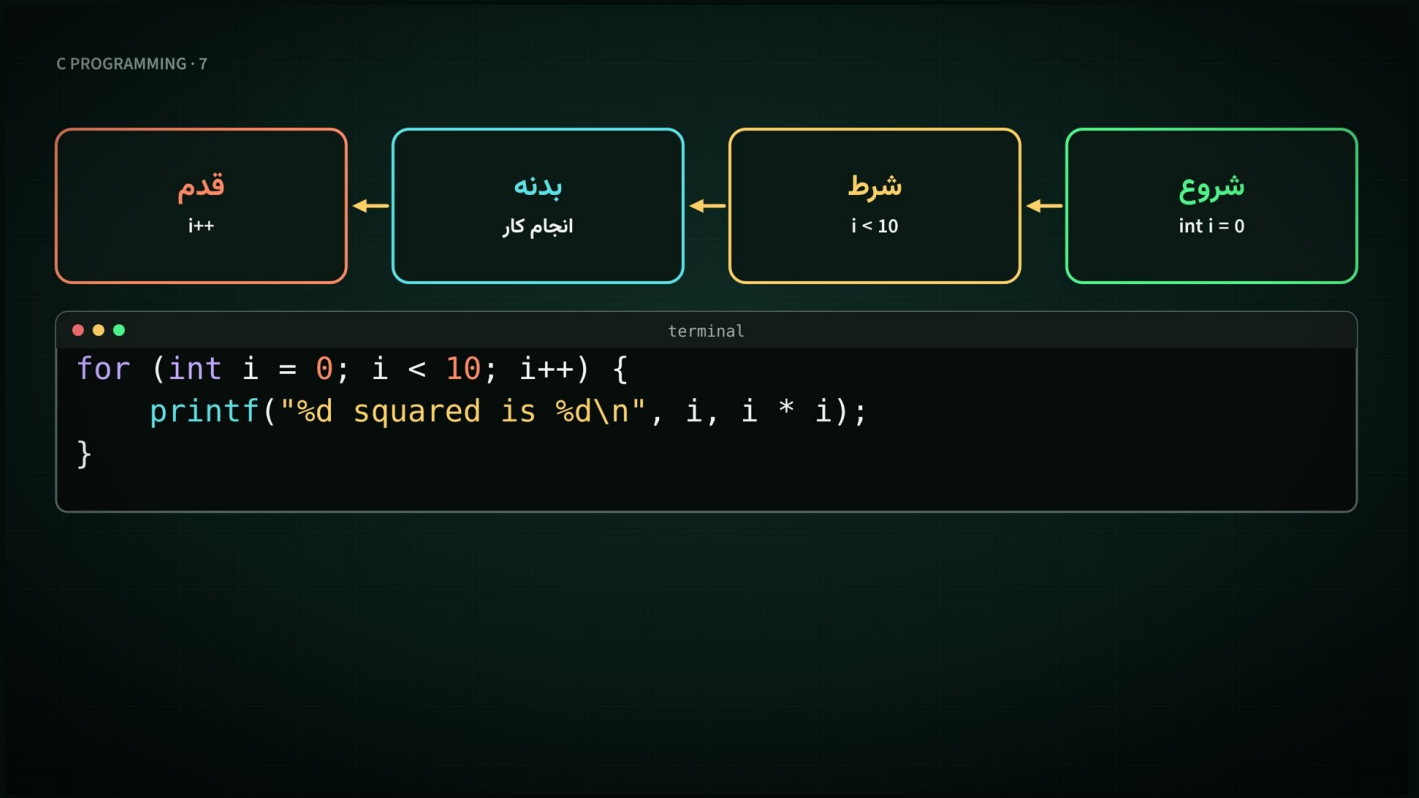
۷.۰۲

حلقهٔ for

شروع — int i = 0

شرط — i < 10

بدنه — انجام کار

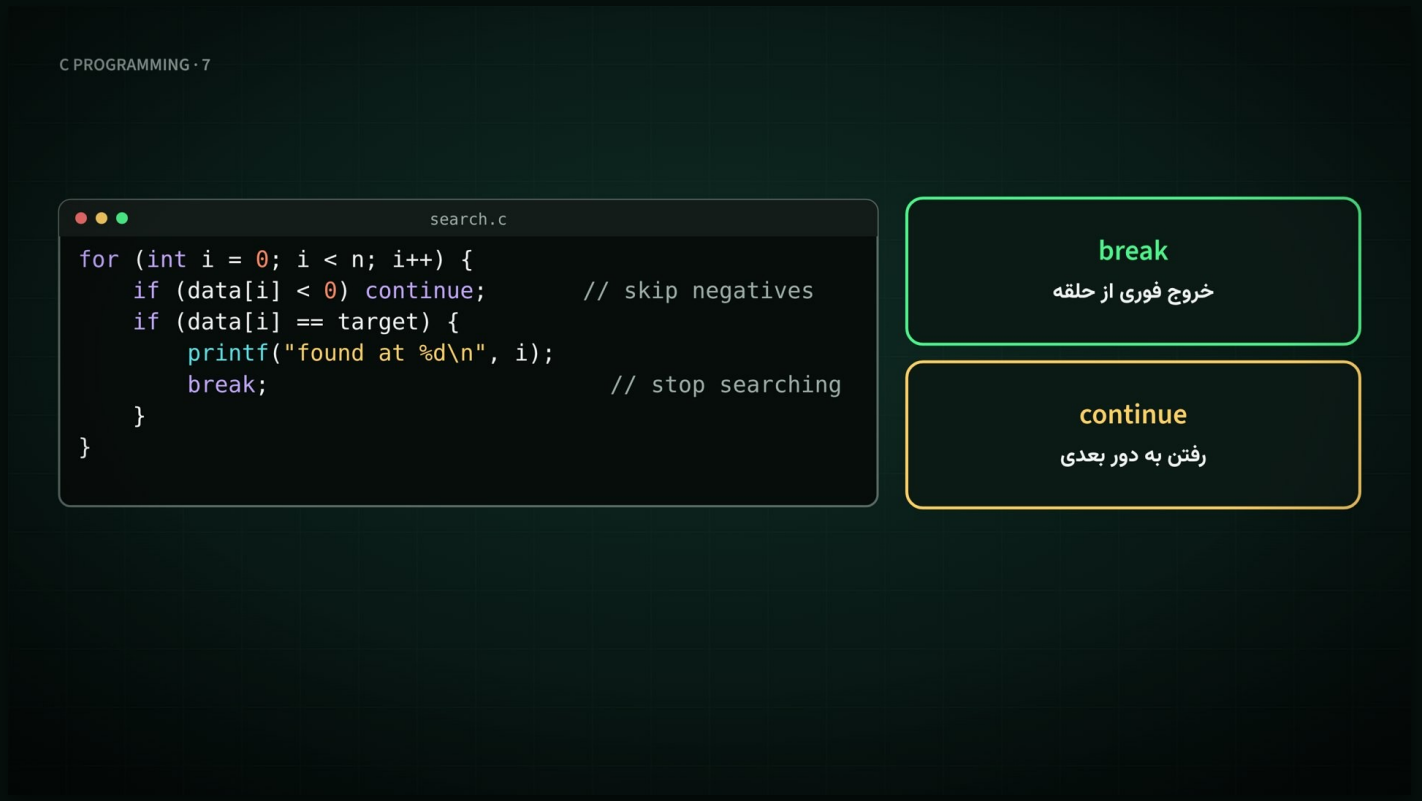


۷.۰۳

break و continue

break — خروج فوری از حلقه

continue — رفتن به دور بعدی



۷.۰۴

حلقه‌های تودرتو

حلقه داخل حلقهٔ دیگر را حلقهٔ تودرتو می‌گویند.
حلقهٔ داخلی برای هر دور حلقهٔ بیرونی به طور ...

حلقه‌های تودرتو جدول رسم می‌کنند، شبکه‌ها و
تصویرها را پردازش می‌کنند، و هر عنصر را با هر ...

C PROGRAMMING · 7

table.c

```c
for (int row = 1; row <= 3; row++) {
    for (int col = 1; col <= 5; col++) {
        printf("%4d", row * col);
    }
    printf("\n");
}
//    1   2   3   4   5
//    2   4   6   8  10
//    3   6   9  12  15
```

۷.۰۵

حلقه‌های بی‌پایان

حلقه‌ای که شرطش هرگز نادرست نمی‌شود، برای همیشه اجرا می‌شود. معمولاً این یک باگ است: فراموش ...

گاهی عمدی است، مثل سروری که منتظر درخواست‌هاست، که به شکل while یک یا for با دو سیمی‌کالن، ...

اگر پروگرام شما در ترمینل گیر کرد، کنترول C را فشار دهید تا متوقف شود.

عمدی ✓

```c
while (1) {
    request = wait_for_request();
    if (request == STOP) break;
    handle(request);
}
```

باگ ×

```c
int i = 0;
while (i < 10) {
    printf("%d", i);
    // forgot i++
}
```

پروگرام در حال اجرا را متوقف می‌کند Ctrl + C

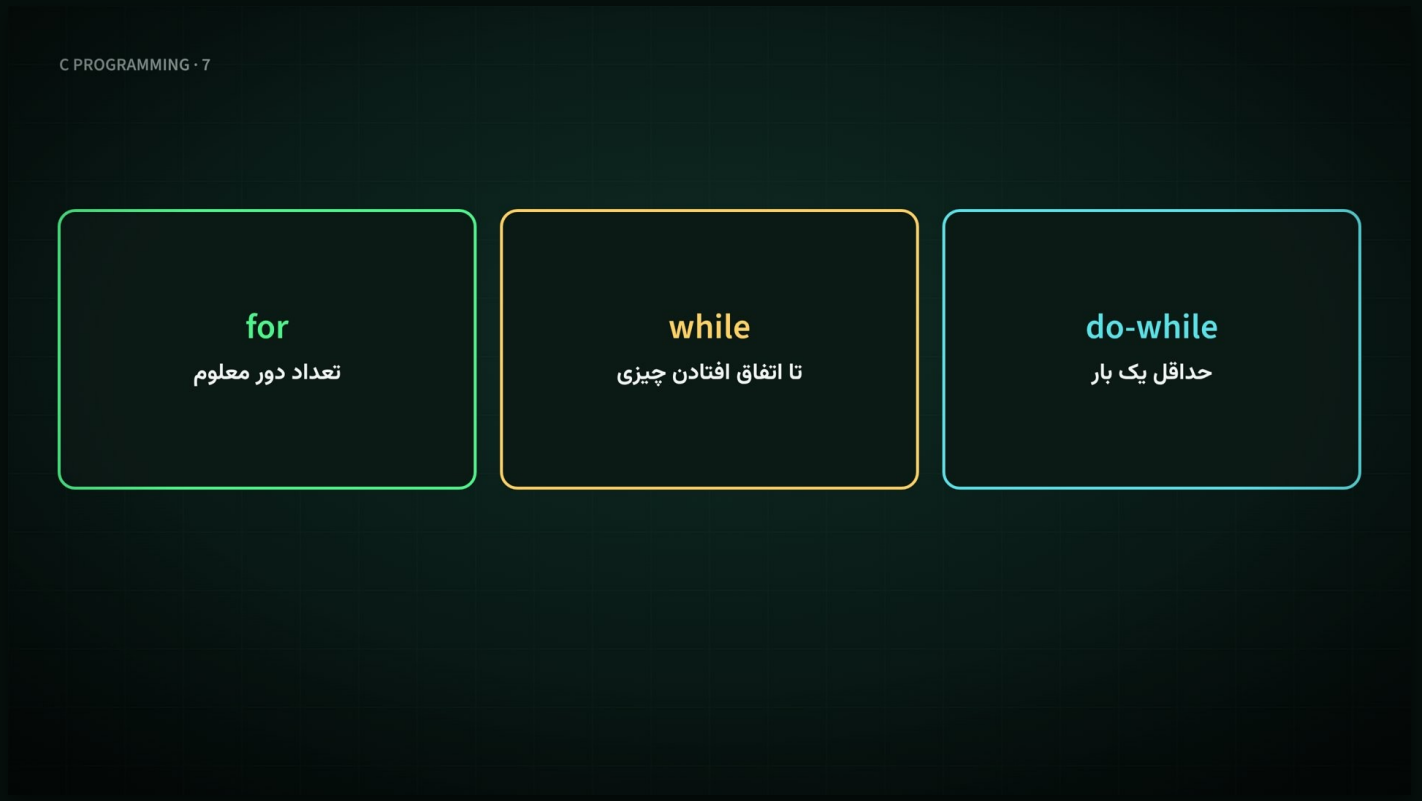
۷.۰۶

از کدام حلقه استفاده کنم؟

for — تعداد دور معلوم

while — تا اتفاق افتادن چیزی

do-while — حداقل یک بار



۷.۰۷

یک کورس رایگان از MyWebUniversity

برنامه‌نویسی C

درس ۸

تابع‌ها، پروتوتایپ‌ها و حوزه

برنامه‌نویسی از بالا به پایین با تابع‌های ساخت خودتان



چرا تابع؟

استفادهٔ دوباره — یک بار بنویس، بارها صدا کن

خوانا — قطعه‌های کوچک نام‌دار

قابل آزمایش — هر کار را جدا بررسی کن



۸.۰۱

تعریف یک تابع

نوع برگشتی — int، double، void ...

پارامترها — ورودی‌ها

return — فرستادن نتیجه

C PROGRAMMING · 8

square.c

```c
//  return type  name    parameters
    int           square (int n) {
        return n * n;            // send back a value
    }

    void greet(const char *name) {   // returns nothing
        printf("Hello, %s!\n", name);
    }

    int r = square(7);   // call: r = 49
    greet("Wahid");
```

نوع برگشتی
int، double، void ...

پارامترها
ورودی‌ها

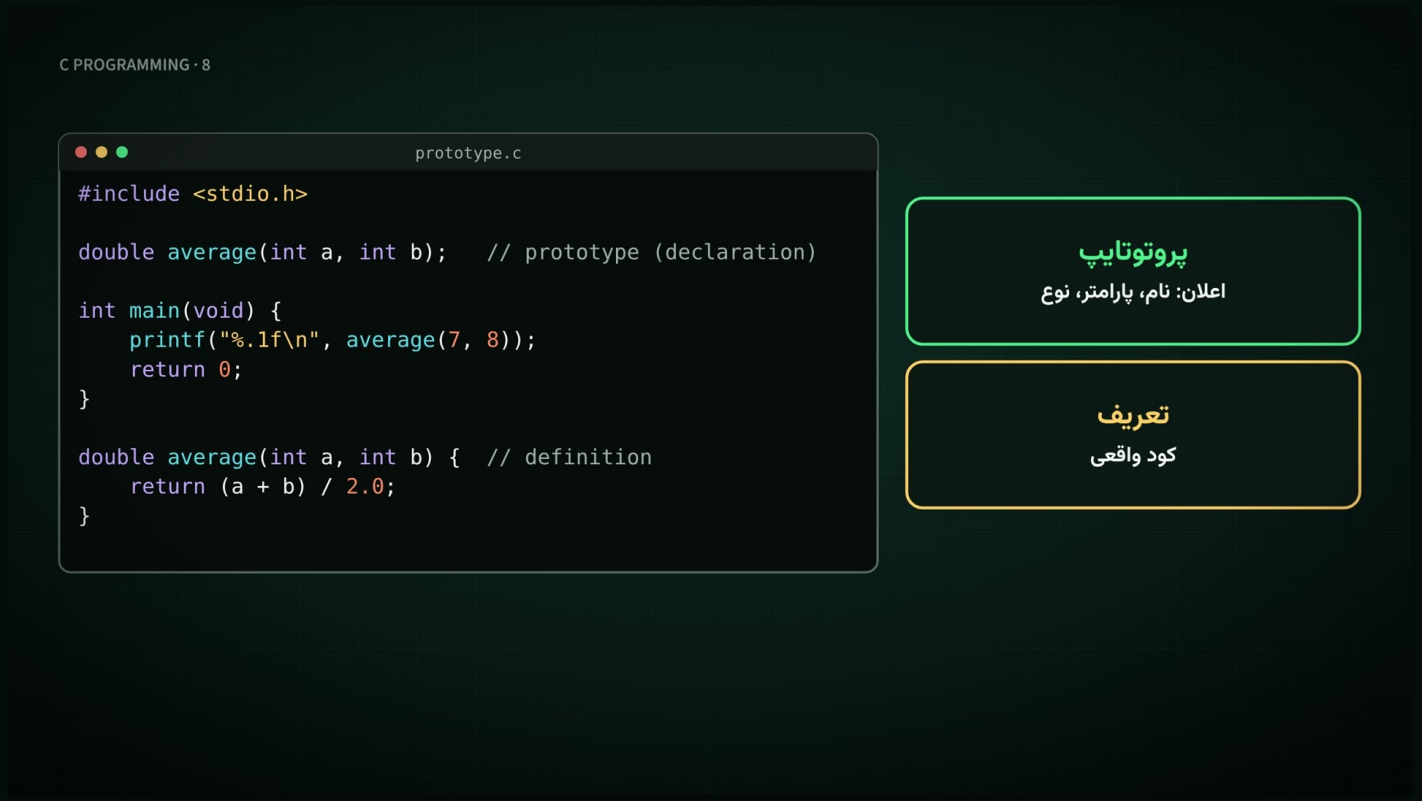
return
فرستادن نتیجه

۸.۰۲

پروتوتایپ‌ها

پروتوتایپ — اعلان: نام، پارامتر، نوع

تعریف — کود واقعی

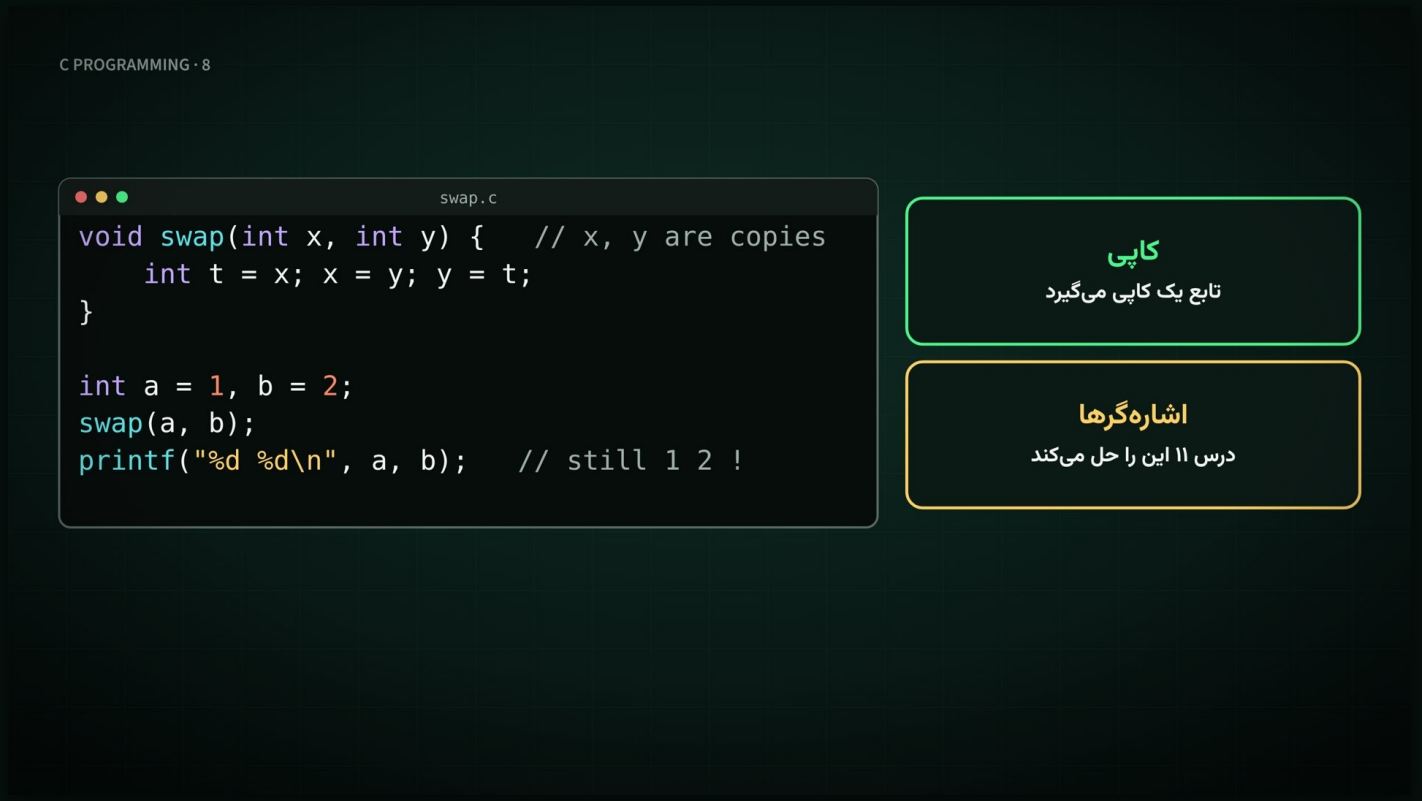


۸.۰۳

فرستادن با مقدار

کاپی — تابع یک کاپی می‌گیرد

اشاره‌گرها — درس ۱۱ این را حل می‌کند



۸.۰۴

حوزه و عمر متغیر

محلی — داخل بلاک خود

سراسری — همه جا — کم استفاده

محلی static — بین صدا کردن‌ها به یاد دارد

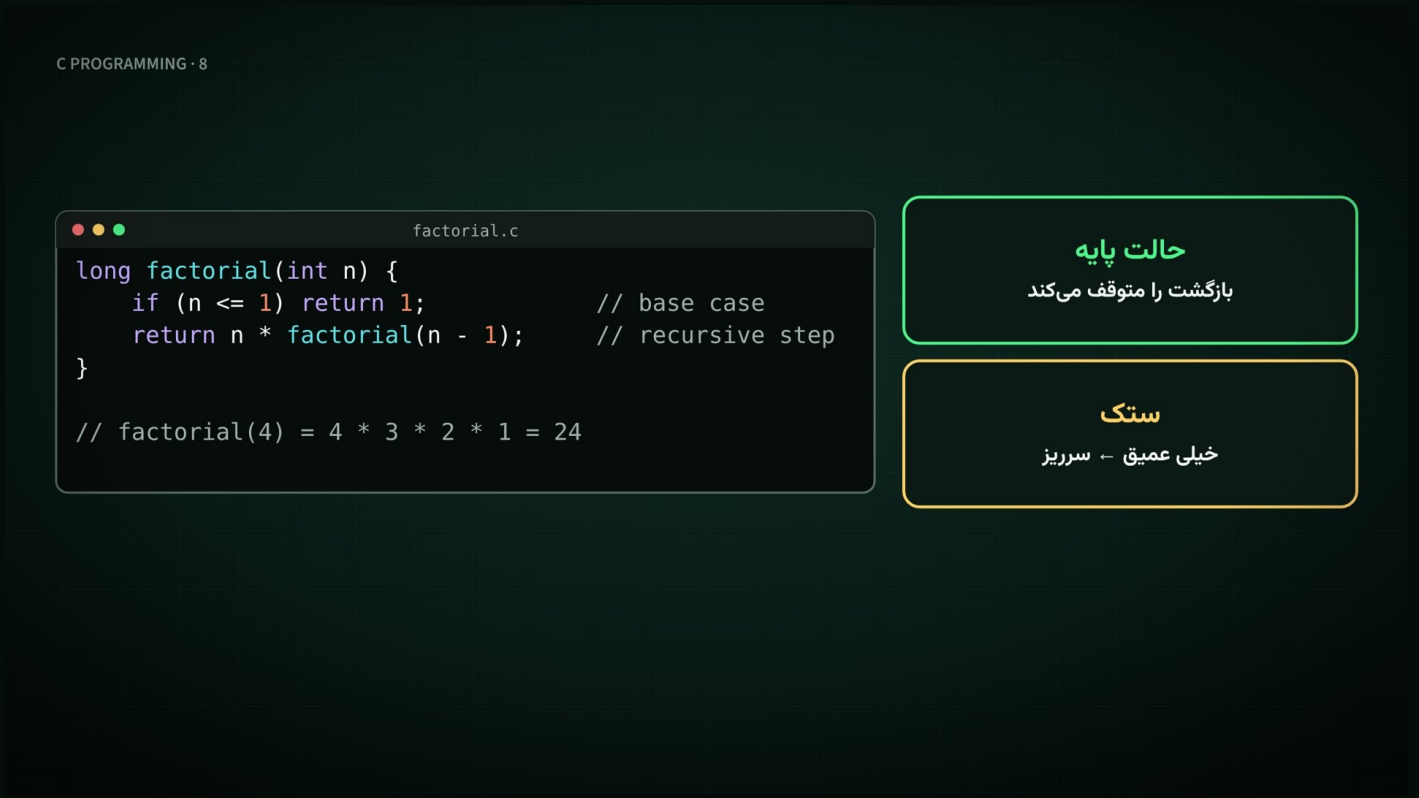


۸.۰۵

بازگشت

حالت پایه — بازگشت را متوقف می‌کند

ستک — خیلی عمیق ← سرریز



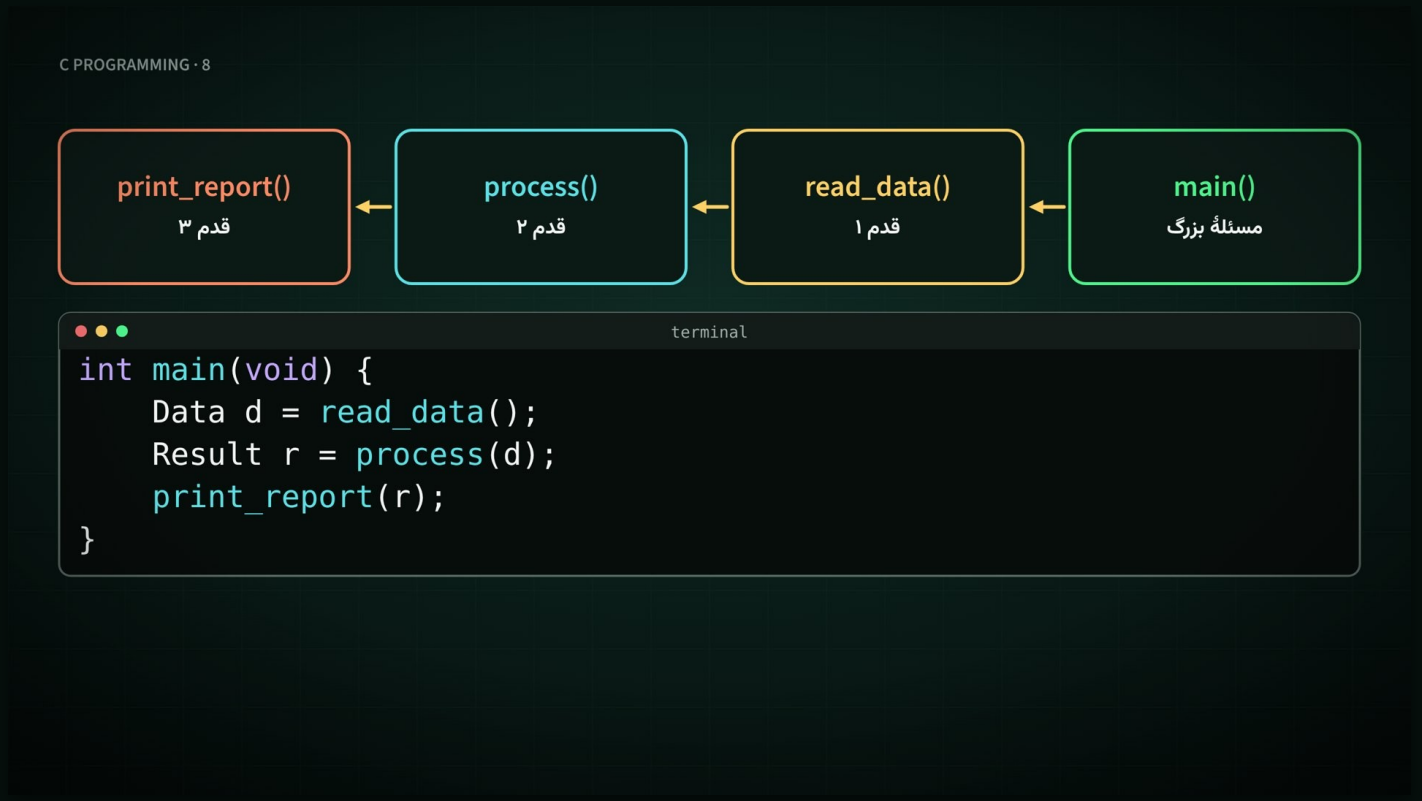
۸.۰۶

طراحی از بالا به پایین

main() — مسئلهٔ بزرگ

read_data() — قدم ۱

process() — قدم ۲



۸.۰۷

یک کورس رایگان از MyWebUniversity

برنامه‌نویسی C

درس ۹

آرایه‌ها و آرایه‌های چندبعدی

ذخیرهٔ چندین مقدار زیر یک نام



آرایه چیست؟

آرایه چندین مقدار از یک نوع را پهلو به پهلو در حافظه، زیر یک نام، ذخیره می‌کند.

به جای پنج متغیر برای پنج نمرهٔ امتحان، یک آرایهٔ پنج‌تایی int می‌نویسید. به هر عنصر با ...

اندکس‌ها از صفر شروع می‌شوند. پس آرایهٔ پنج عنصری اندکس‌های صفر تا چهار دارد.

C PROGRAMMING · 9

main.c

```c
int scores[5] = {90, 85, 77, 64, 98};
scores[2] = 80;            // change the third element
```



۹.۰۱

آرایه‌ها و حلقه‌ها

آرایه‌ها و حلقه‌های for بهترین دوستان هستند.
شمارندهٔ حلقه از صفر تا طول آرایه، بدون خود ...

برای گرفتن تعداد عناصر، اندازهٔ تمام آرایه را بر اندازهٔ یک عنصر تقسیم کنید. آن را در یک ...

C PROGRAMMING · 9

average.c

```c
int scores[] = {90, 85, 77, 64, 98};
int n = sizeof(scores) / sizeof(scores[0]);   // 5

int sum = 0;
for (int i = 0; i < n; i++) {
    sum += scores[i];
}
printf("average = %.1f\n", (double)sum / n);   // 82.8
```

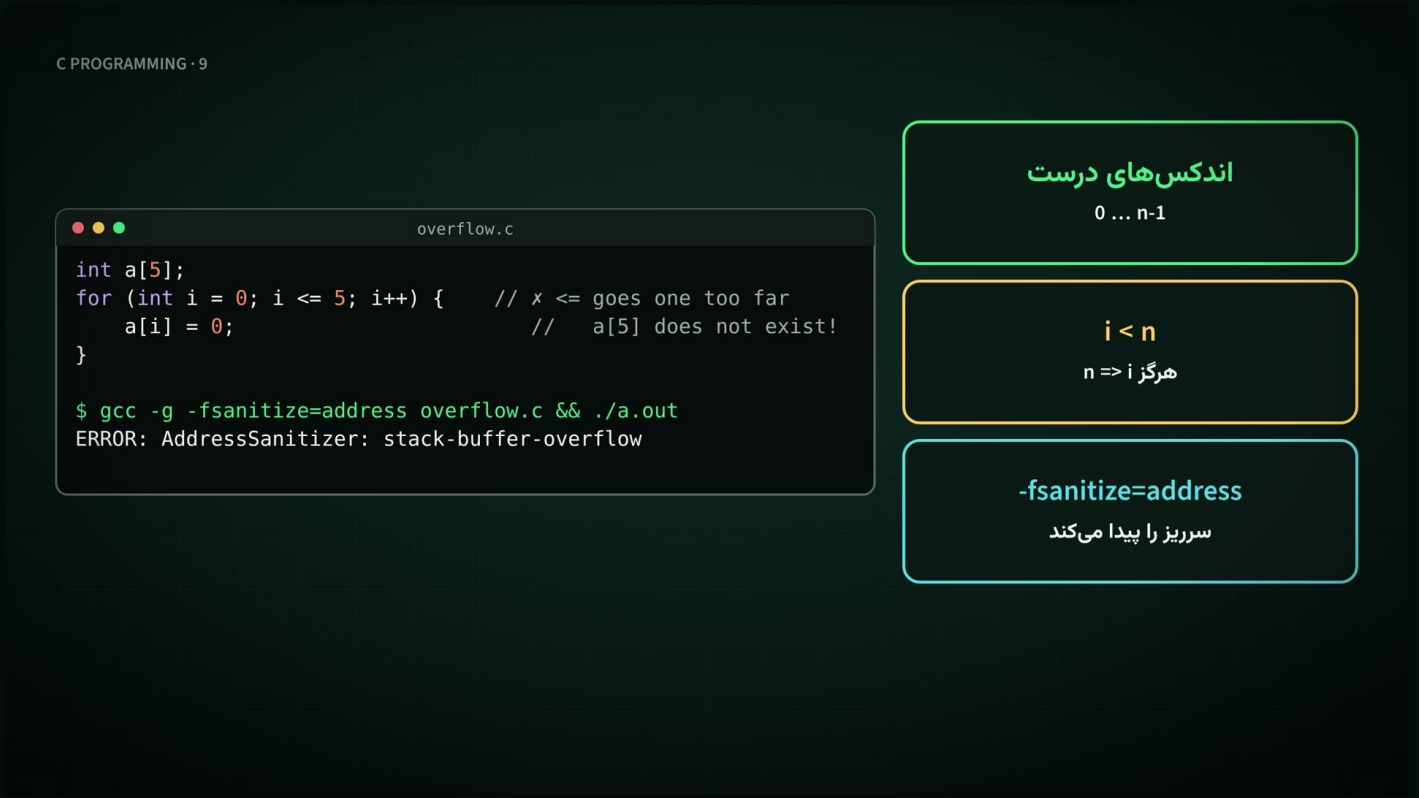
۹.۰۲

بدون بررسی مرز!

اندکس‌های درست — 0 … n-1

i < n — هرگز i <= n

-fsanitize=address — سرریز را پیدا می‌کند

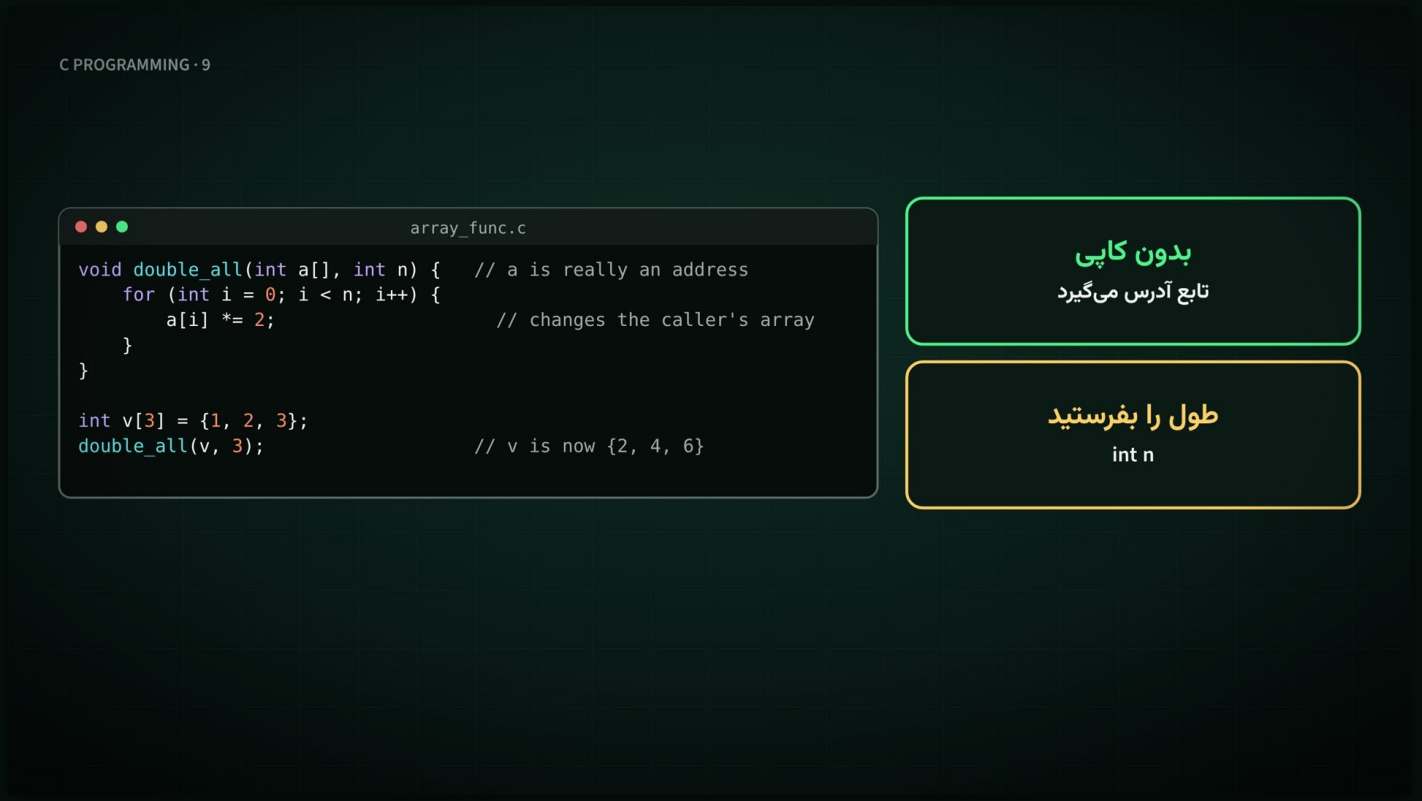


۹.۰۳

فرستادن آرایه به تابع

بدون کاپی — تابع آدرس می‌گیرد

طول را بفرستید — int n



۹.۰۴

آرایه‌های دوبعدی

m[row][col] — اول سطر

ترتیب سطری — سطرها پشت سر هم



۹.۰۵

ابعاد بیشتر و کاربردها

بازی‌ها — شطرنج، دوز

جدول‌ها — سطر × ستون

تصویرها — pixels[y][x]



۹.۰۶

جستجو و مرتب‌سازی

دو کار کلاسیک روی آرایه‌ها جستجو و مرتب‌سازی است. جستجوی خطی هر عنصر را بررسی می‌کند؛ ساده ...

اگر آرایه مرتب باشد، جستجوی دودویی در هر قدم محدوده را نصف می‌کند، پس یک عنصر را بین یک ...

کتابخانهٔ C qsort را برای مرتب‌سازی و bsearch را برای جستجوی دودویی دارد. qsort را در ...



۹.۰۷

یک کورس رایگان از MyWebUniversity

برنامه‌نویسی C

درس ۱۰

رشته‌ها در C

آرایه‌های char، پایان‌دهندهٔ NULL و فایل string

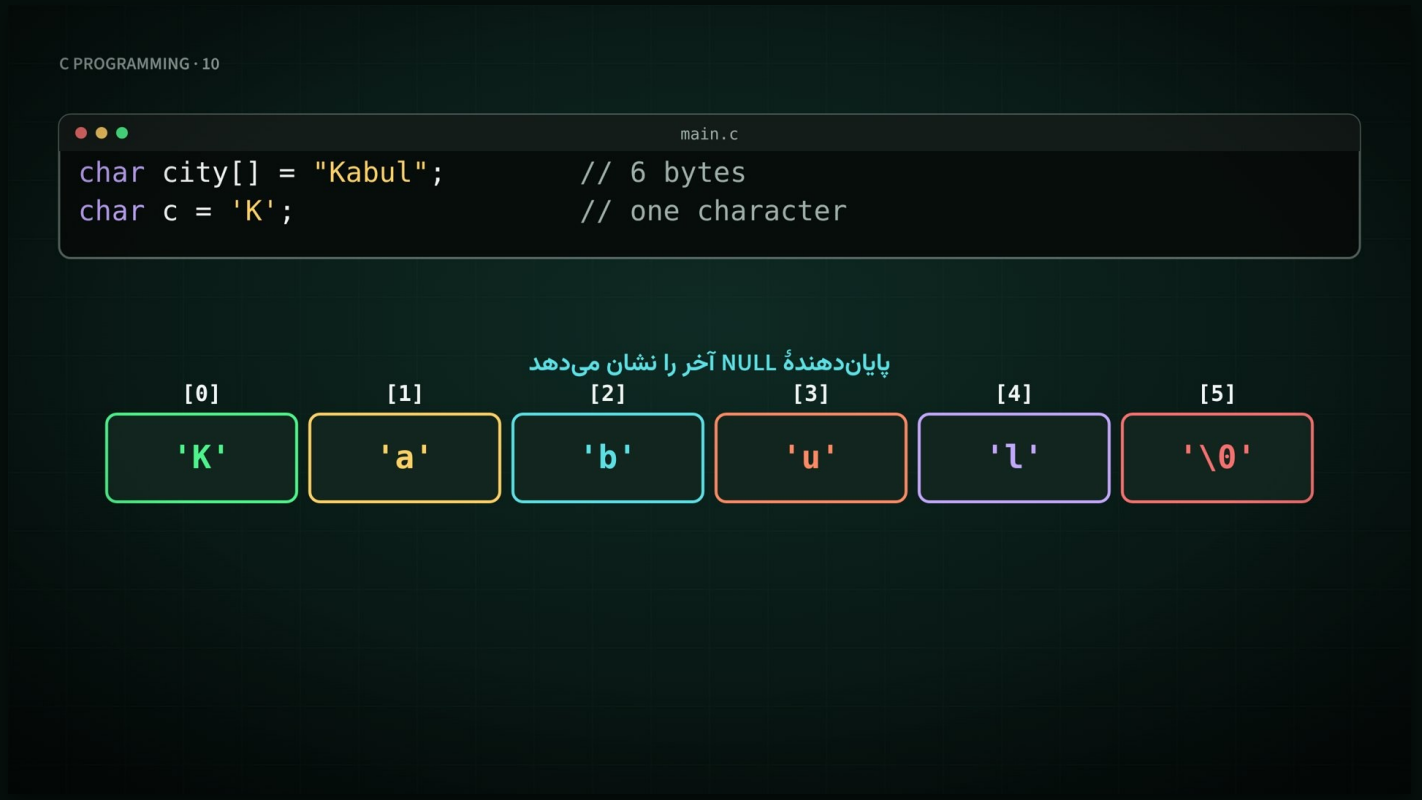


رشته یک آرایهٔ char است

C نوع ویژه‌ای برای رشته ندارد. رشته آرایه‌ای از حروف است که با یک حرف ویژه، backslash صفر، ...

کلمهٔ کابل شش بایت لازم دارد: پنج حرف و پایان‌دهنده. هر تابع رشته به همان صفر تکیه می‌کند ...

نقل قول یکه یک حرف می‌سازد. نقل قول جفت یک رشته می‌سازد و کمپایلر پایان‌دهنده را برای شما ...



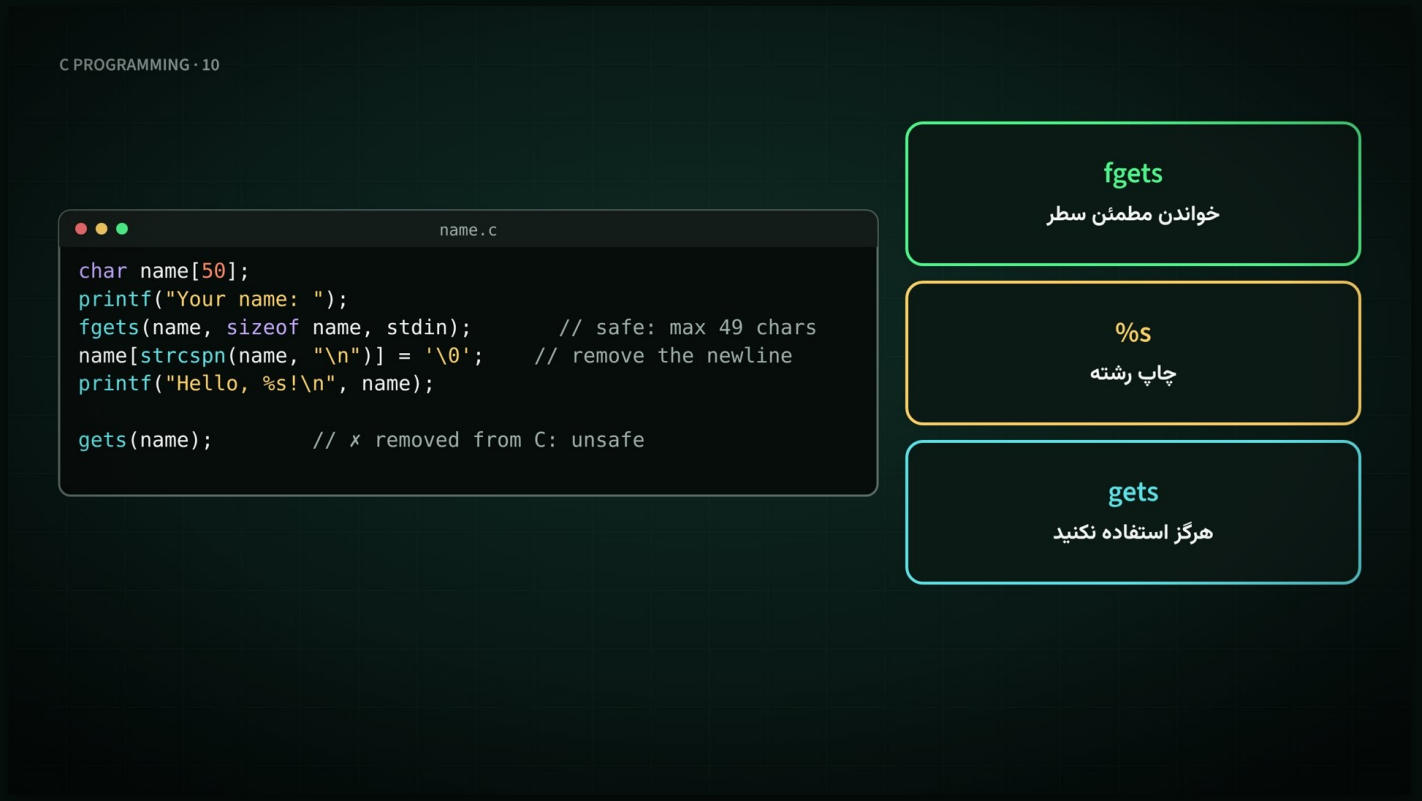
۱۰.۰۱

خواندن و چاپ رشته‌ها

fgets — خواندن مطمئن سطر

s% — چاپ رشته

gets — هرگز استفاده نکنید



۱۰.۰۲

تابع‌های فایل string

فایل هِدر string تابع‌هایی برای کار با رشته‌ها می‌دهد. strlen طول را بدون پایان‌دهنده ...

strcpy یک رشته را در دیگری کاپی می‌کند، strcat به آخر اضافه می‌کند، و strcmp دو ...

نمی‌توانید رشته‌ها را با دو علامت مساوی مقایسه کنید یا با مساوی مقدار بدهید؛ این‌ها آدرس‌ها ...

C PROGRAMMING · 10

تابع	مثال	کار
strlen(s)	strlen("Kabul") → 5	طول
strcpy(d, s)	strcpy(dst, "Hi")	کاپی
strcat(d, s)	strcat(dst, "!")	اضافه به آخر
strcmp(a, b)	strcmp("a","b") → <0	مقایسه (۰ = برابر)
strchr(s, c)	strchr(s, '@')	پیدا کردن حرف

۱۰.۰۳

کار مطمئن‌تر با رشته‌ها

snprintf — هرگز از اندازه بیشتر نمی‌نویسد

اندازه + ۱ — جای '\0'



۱۰.۰۴

کار حرف به حرف

چون رشته یک آرایه است، می‌توانید تا رسیدن به پایان‌دهنده در آن حلقه بزنید.

فایل هِدر ctype به دسته‌بندی حروف کمک می‌کند: ایز دیجیت، ایز اَلفا، ایز اِسپیس، و تو اَپر ...

C PROGRAMMING · 10

upper.c

```c
#include <ctype.h>

void to_upper(char *s) {
    for (int i = 0; s[i] != '\0'; i++) {
        s[i] = toupper((unsigned char)s[i]);
    }
}

char w[] = "kabul";
to_upper(w);          // w is now "KABUL"
```

۱۰.۰۵

آرایه در برابر لیترال رشته

بین یک آرایهٔ char و یک اشاره‌گر به لیترال رشته تفاوت مهمی وجود دارد.

آرایهٔ char کاپی خود را از متن دارد، پس می‌توانید آن را تغییر دهید. اشاره‌گر به لیترال به ...



۱۰.۰۶

یک کورس رایگان از MyWebUniversity

برنامه‌نویسی C

درس ۱۱

اشاره‌گرها

آدرس‌ها، دسترسی غیرمستقیم، حساب اشاره‌گر و اشاره‌گر تابع



هر متغیر یک آدرس دارد

حافظه یک ردیف طولانی از بایت‌های شماره‌دار است. شمارهٔ یک بایت آدرس آن است، درست مثل نمبر ...

عملگر اَمپرسَند آدرس یک متغیر را به شما می‌دهد. می‌توانید آن را با p% چاپ کنید.

اشاره‌گر به سادگی متغیری است که یک آدرس را ذخیره می‌کند.

C PROGRAMMING · 11

main.c

```c
int age = 42;
int *p = &age;            // p holds the address of age
printf("%p\n", (void*)p);  // 0x7ffd1000
```



۱۱.۰۱

اعلان و دسترسی غیرمستقیم

&x — آدرس x

p — int *p اشاره‌گر است

*p — مقداری که p نشان می‌دهد

C PROGRAMMING · 11

deref.c

```c
int age = 42;
int *p = &age;     // declaration: p is a pointer to int

*p = 50;           // dereference: change age through p
printf("%d\n", age);   // 50
printf("%d\n", *p);    // 50
```

&x
آدرس x

int *p
p اشاره‌گر است

***p**
مقداری که p نشان می‌دهد

۱۱.۰۲

فرستادن با مرجع: swap کار می‌کند!

&a, &b — آدرس‌ها را بفرست

*x = *y — تغییر اصلی‌ها

C PROGRAMMING · 11

swap.c

```c
void swap(int *x, int *y) {
    int t = *x;
    *x = *y;
    *y = t;
}

int a = 1, b = 2;
swap(&a, &b);
printf("%d %d\n", a, b);   // 2 1  ✓
```

&a, &b
آدرس‌ها را بفرست

***x = *y**
تغییر اصلی‌ها

۱۱.۰۳

اشاره‌گرها و آرایه‌ها

نام آرایه به اشاره‌گری به عنصر اول آن تبدیل می‌شود. پس a و &a[0] یکی هستند.

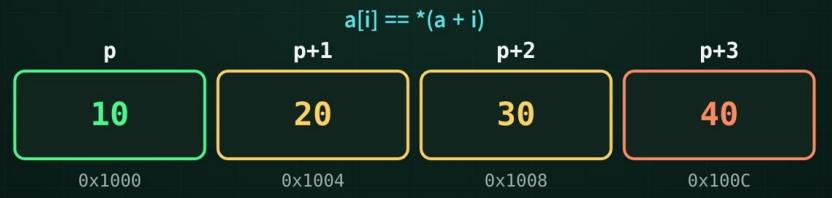
حساب اشاره‌گر با عناصر کامل حرکت می‌کند، نه بایت: p + 1 به int بعدی اشاره می‌کند، چهار ...

در واقع، a[i] به معنای *(a + i)) تعریف شده. به همین دلیل آرایه‌ها و ...

C PROGRAMMING · 11

main.c

```c
int a[4] = {10, 20, 30, 40};
int *p = a;                    // same as &a[0]
printf("%d\n", *(p + 2));   // 30, same as a[2]
```



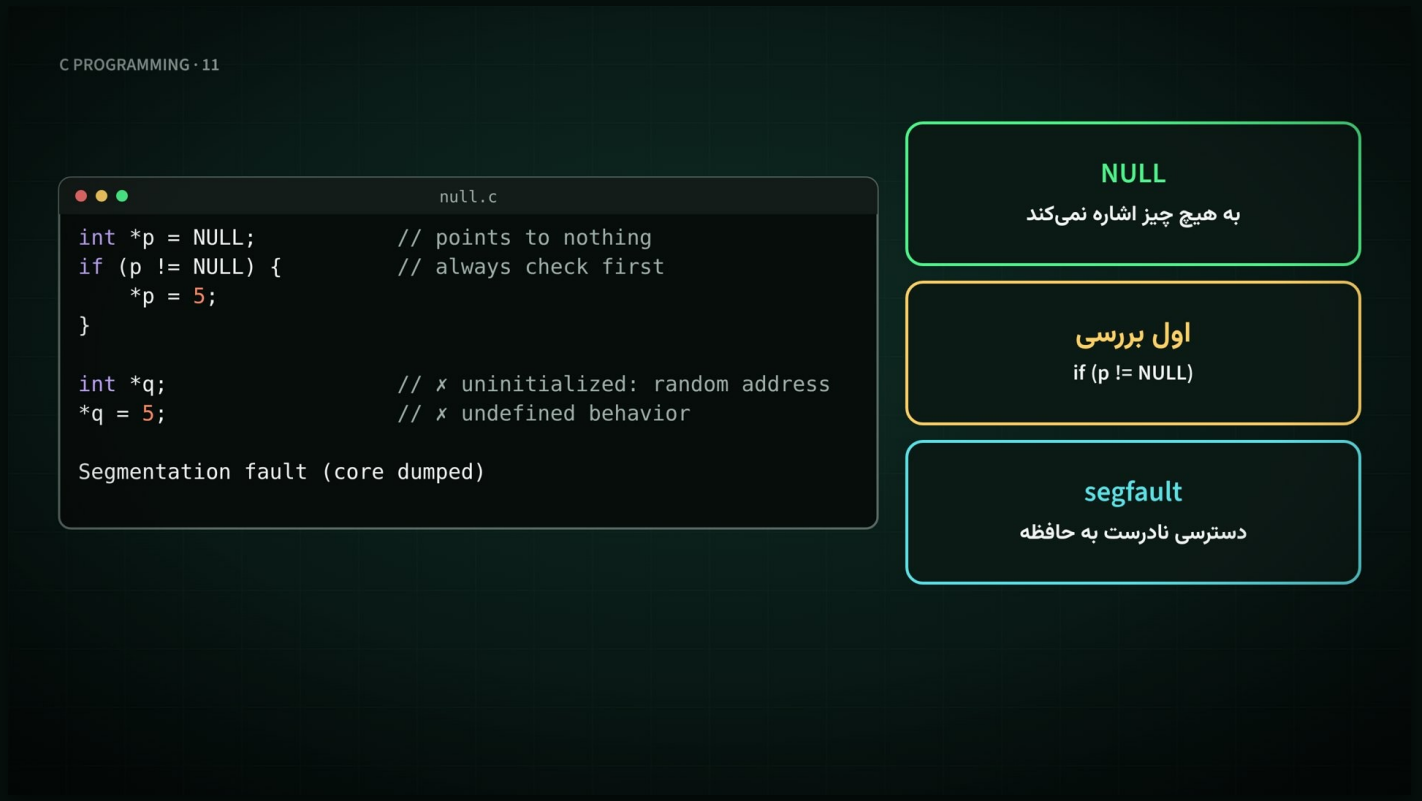
۱۱.۰۴

NULL و اشاره‌گرهای خطرناک

NULL — به هیچ چیز اشاره نمی‌کند

اول بررسی — if (p != NULL)

segfault — دسترسی نادرست به حافظه

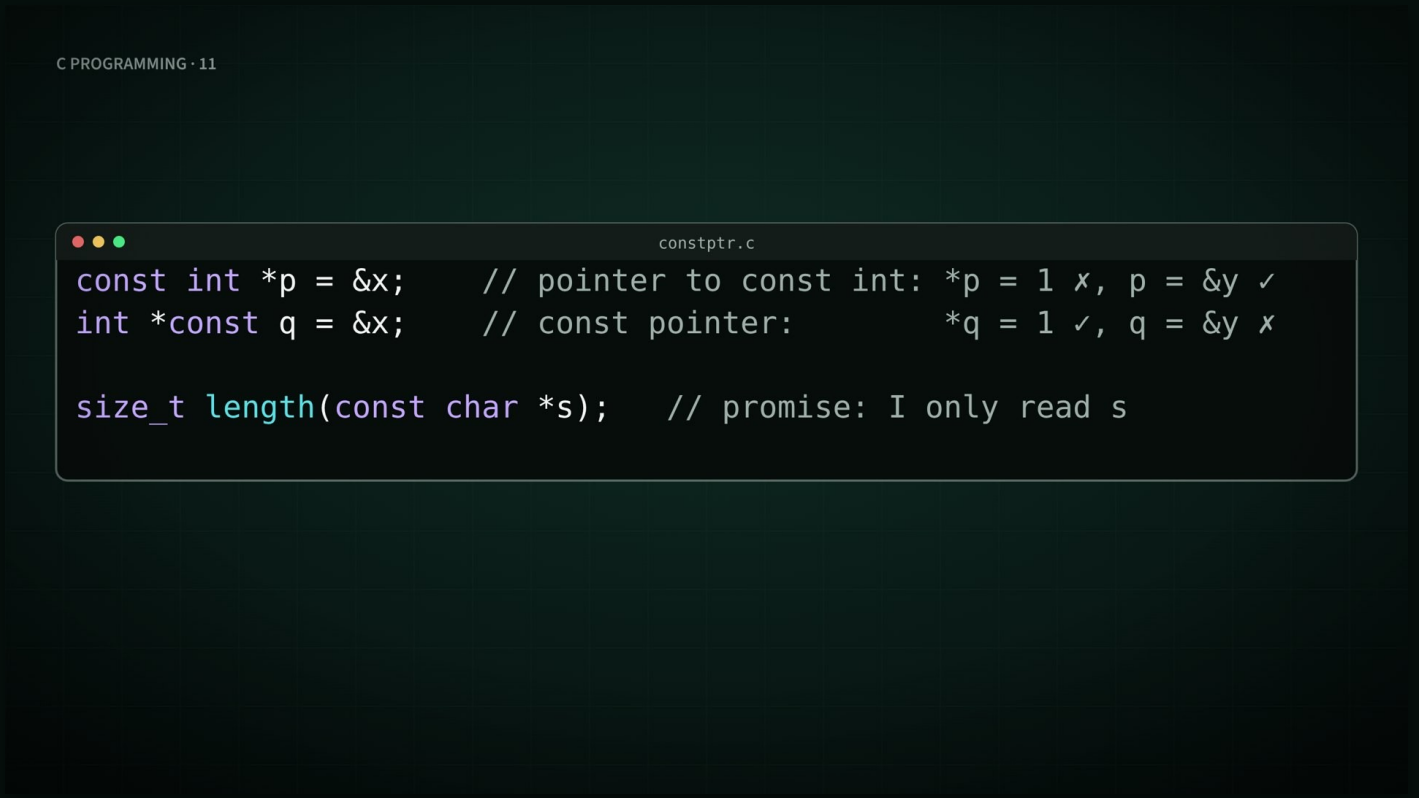


۱۱.۰۵

const با اشاره‌گرها

const و اشاره‌گرها به دو شکل ترکیب می‌شوند.
اشاره‌گر به const یعنی نمی‌توانید مقدار را از ...

اشاره‌گر const یعنی خود اشاره‌گر نمی‌تواند به جای دیگری اشاره کند. اعلان را از راست به چپ ...



۱۱.۰۶

اشاره‌گر به اشاره‌گر و اشاره‌گر تابع

یک اشاره‌گر می‌تواند به اشاره‌گر دیگری اشاره کند: int ستاره ستاره. این را با argv، ...

اشاره‌گر تابع آدرس یک تابع را ذخیره می‌کند، پس می‌توانید رفتار را به عنوان آرگومان بفرستید. ...

C PROGRAMMING · 11

fptr.c

```c
int main(int argc, char **argv);    // argv: pointer to pointers

int add(int a, int b) { return a + b; }
int mul(int a, int b) { return a * b; }

int (*op)(int, int) = add;          // function pointer
printf("%d\n", op(3, 4));   // 7
op = mul;
printf("%d\n", op(3, 4));   // 12
```

۱۱.۰۷

یک کورس رایگان از MyWebUniversity

برنامه‌نویسی C

درس ۱۲

حافظهٔ پویا و نشت حافظه

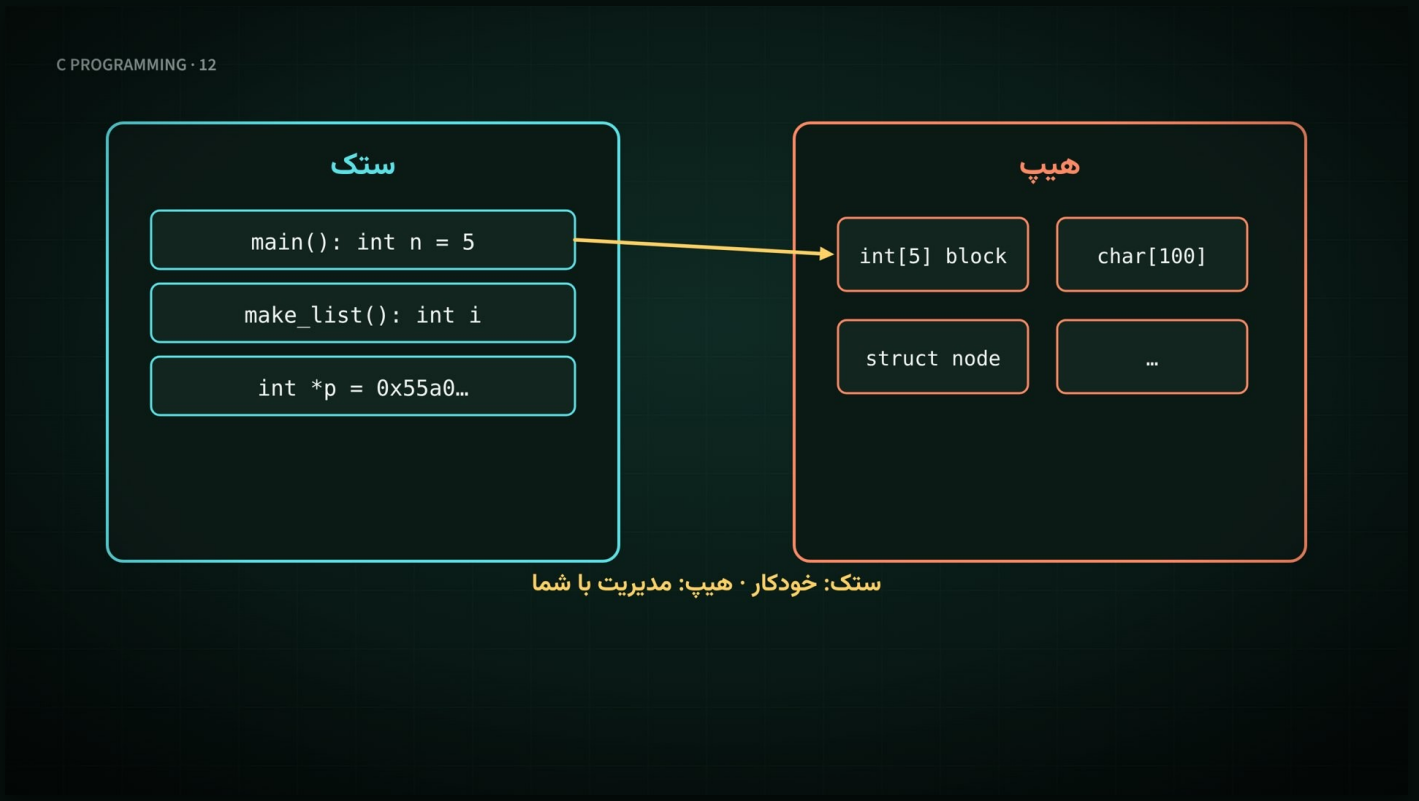
ستک در برابر هیپ، malloc، free، و پیدا کردن نشت



ستک و هیپ

یک پروگرام C در حال اجرا دو بخش اصلی حافظه را استفاده می‌کند. ستک متغیرهای محلی را نگه ...

هیپ یک حوض بزرگ حافظه است که خودتان درخواست و آزاد می‌کنید. از آن وقتی استفاده کنید که ...



۱۲.۰۱

malloc و free

malloc — رزرو بایت روی هیپ

بررسی NULL — رزرو می‌تواند ناکام شود

free — هر malloc یک free

C PROGRAMMING · 12

dynamic.c

```c
#include <stdlib.h>

int n;
scanf("%d", &n);                        // size known only at run time
int *a = malloc(n * sizeof *a);
if (a == NULL) {
    fprintf(stderr, "out of memory\n");
    return 1;
}
for (int i = 0; i < n; i++) a[i] = i * i;
free(a);                                // give it back
a = NULL;                               // avoid a dangling pointer
```

malloc
رزرو بایت روی هیپ

بررسی NULL
رزرو می‌تواند ناکام شود

free
هر malloc یک free

۱۲.۰۲

calloc و realloc

calloc مثل malloc است اما حافظه را با صفر پر می‌کند. وقتی به حالت شروع پاک ضرورت دارید از آن ...

realloc اندازهٔ یک بلاک را تغییر می‌دهد، مثلاً برای بزرگ کردن یک آرایه وقتی دیتای بیشتر ...

C PROGRAMMING · 12

grow.c

```c
int *a = calloc(10, sizeof *a);          // 10 zeros

int *bigger = realloc(a, 20 * sizeof *a); // grow to 20
if (bigger == NULL) {
    free(a);                             // old block still valid
    return 1;
}
a = bigger;
free(a);
```

۱۲.۰۳

نشت حافظه

نشت حافظه وقتی اتفاق می‌افتد که حافظه رزرو کنید و آخرین اشاره‌گر به آن را بدون صدا کردن free ...

در یک پروگرام کوتاه شاید مهم نباشد. اما سروری که ماه‌ها اجرا می‌شود و در هر درخواست کمی نشت ...

C PROGRAMMING · 12

leak.c

```c
void handle_request(void) {
    char *buf = malloc(1024);
    if (!buf) return;
    /* ... use buf ... */
    if (error) return;          // ✗ leak: returned without free
    free(buf);
}

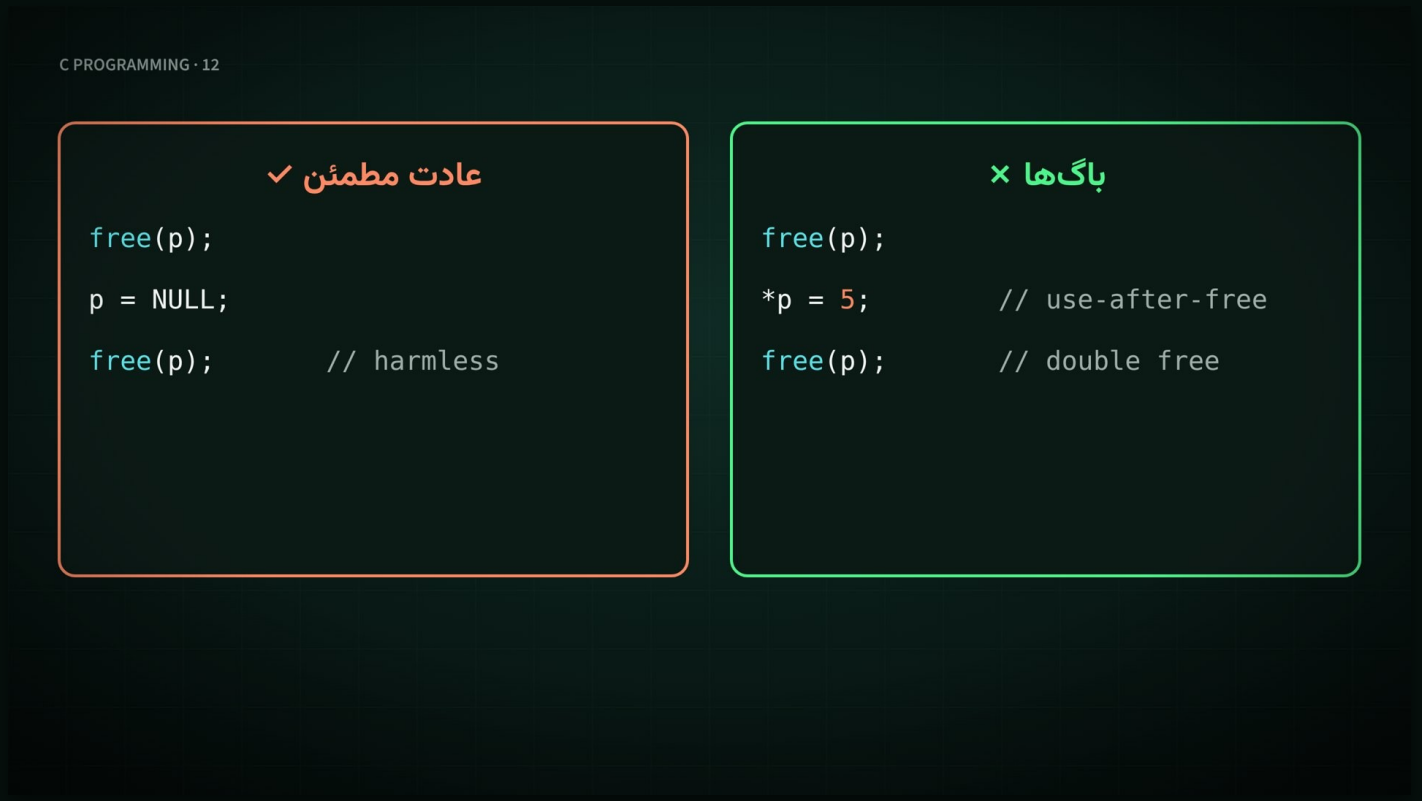
// 1 KB × 1,000,000 requests = 1 GB lost
```

۱۲.۰۴

اشاره‌گرهای آویزان و free دوباره

بعد از free، اشاره‌گر هنوز آدرس قدیم را دارد، اما حافظه دیگر مال شما نیست. استفاده از آن را ...

دو بار صدا کردن free روی یک اشاره‌گر، دوباره است که می‌تواند هیپ را خراب کند. NULL کردن ...

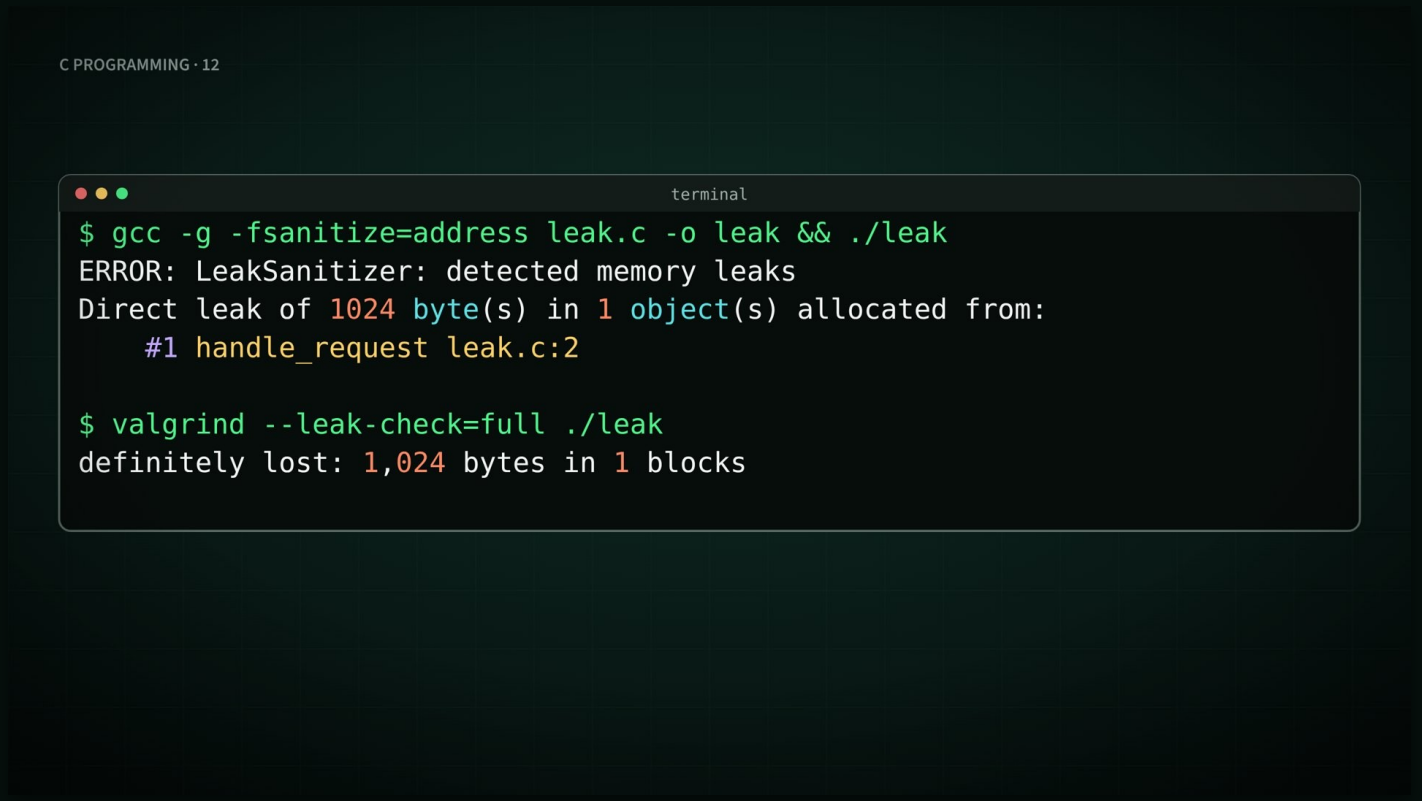


۱۲.۰۵

پیدا کردن باگ‌های حافظه

لازم نیست این باگ‌ها را دستی پیدا کنید. با -fsanitize=address کمپایل کنید، و ...

در Linux، Valgrind پروگرام شما را اجرا می‌کند و هر بایتی را که هرگز آزاد نشده گزارش ...



۱۲.۰۶

یک کورس رایگان از MyWebUniversity

برنامه‌نویسی C

درس ۱۳

struct، union، enum و typedef

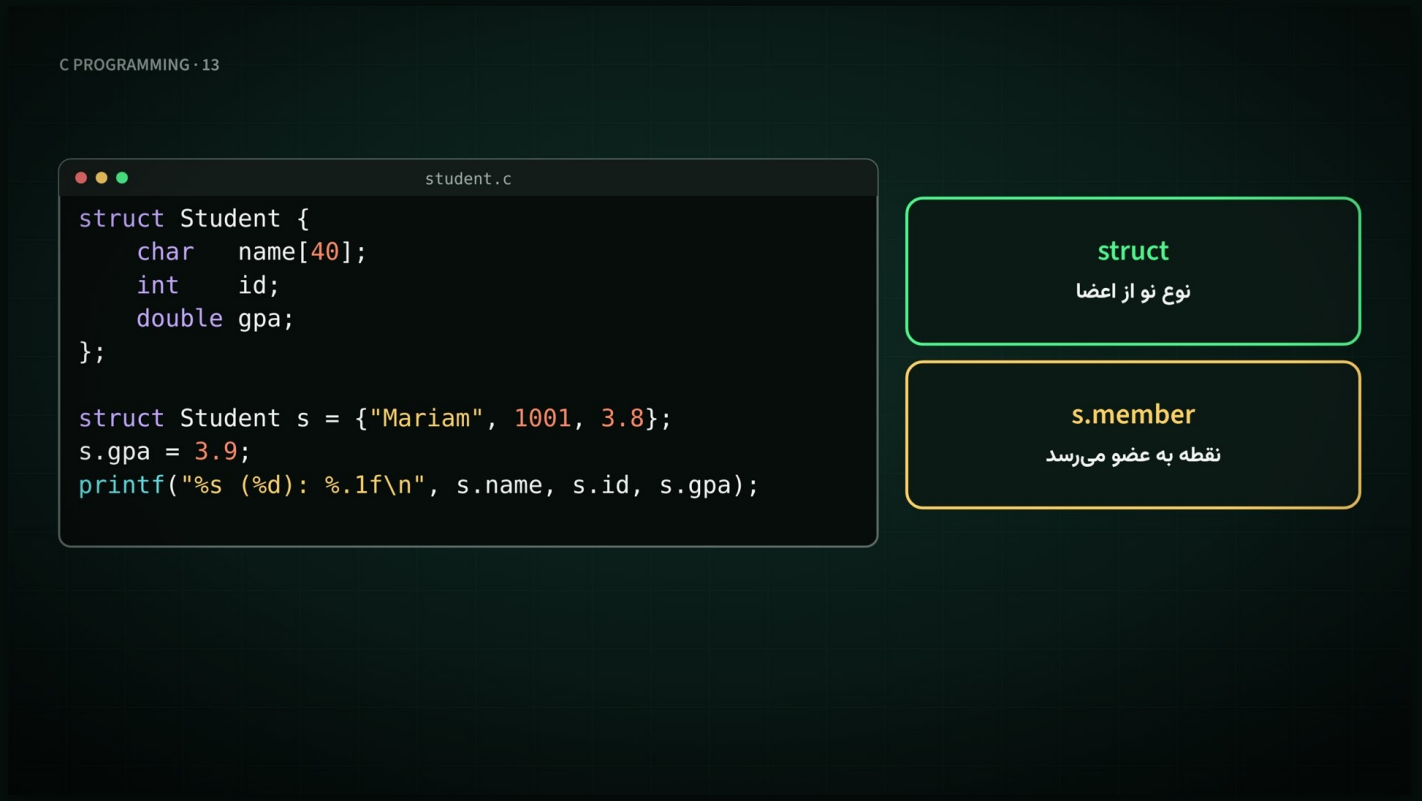
ساختن نوع‌های دیتای خودتان



struct: جمع کردن دیتای مرتبط

struct — نوع نو از اعضا

s.member — نقطه به عضو می‌رسد



۱۳.۰۱

typedef: نام‌های کوتاه‌تر

هر بار نوشتن struct استودنت طولانی است. typedef به یک نوع نام نو می‌دهد، پس می‌توانید به ...

typedef همچنین برای دادن نام‌های با معنا به نوع‌های اصلی استفاده می‌شود، مثل typedef ...

C PROGRAMMING · 13

typedef.c

```c
typedef struct {
    double x, y;
} Point;

Point a = {0, 0}, b = {3, 4};

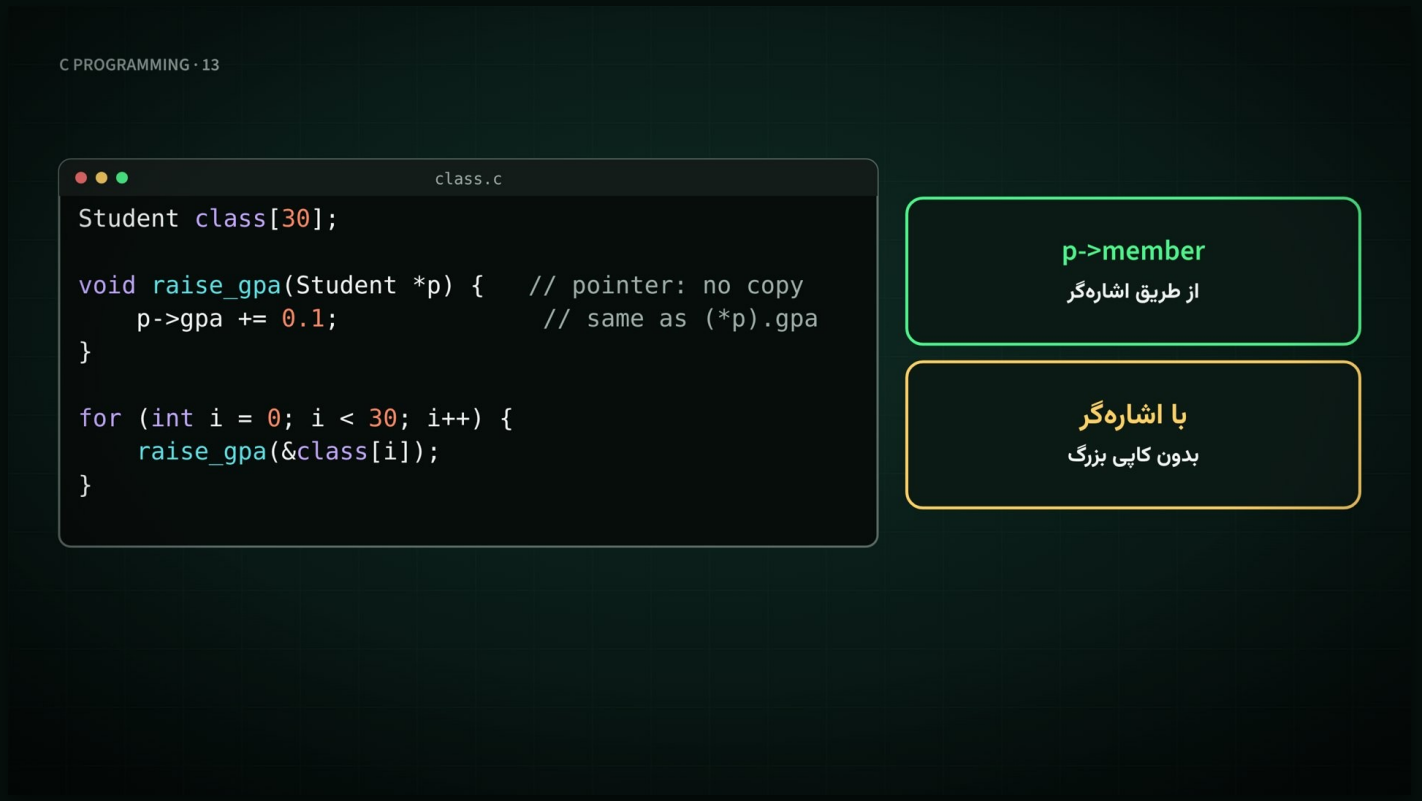
typedef unsigned char byte;
byte pixel = 255;
```

۱۳.۰۲

آرایهٔ structها و تیر

p->member — از طریق اشاره‌گر

با اشاره‌گر — بدون کاپی بزرگ



۱۳.۰۳

چیدمان حافظه و پَدینگ

اعضای یک struct به ترتیب ذخیره می‌شوند، اما کمپایلر شاید بایت‌های خالی پَدینگ اضافه کند تا ...

به همین دلیل sizeof یک struct می‌تواند از مجموع اعضای آن بزرگ‌تر باشد. مرتب کردن اعضا از ...

C PROGRAMMING · 13

main.c

```
struct A { char c; int n; char d; };   // sizeof = 12
struct B { int n; char c; char d; };   // sizeof = 8
```



۱۳.۰۴

union: اشتراک حافظه

union — اعضا حافظه را مشترک دارند

اندازه — بزرگ‌ترین عضو

برچسب — به یاد داشتن محتوا

C PROGRAMMING · 13

union.c

```c
typedef enum { INT, REAL, TEXT } Kind;

typedef struct {
    Kind kind;              // which member is valid
    union {
        int    i;
        double d;
        char   s[16];
    } as;                   // 16 bytes, shared
} Value;

Value v = { .kind = REAL, .as.d = 3.14 };
```

union
اعضا حافظه را مشترک دارند

اندازه
بزرگ‌ترین عضو

برچسب
به یاد داشتن محتوا

۱۳.۰۵

enum: ثابت‌های نام‌دار

enum مجموعه‌ای از ثابت‌های عدد صحیح نام‌دار را تعریف می‌کند. به جای به یاد داشتن اینکه دو ...

به طور پیش‌فرض مقدارها از صفر شروع می‌شوند و بالا می‌روند، اما می‌توانید خودتان انتخاب ...

C PROGRAMMING · 13

enum.c

```c
enum Light { RED, GREEN, YELLOW };   // 0, 1, 2
enum Http  { OK = 200, NOT_FOUND = 404 };

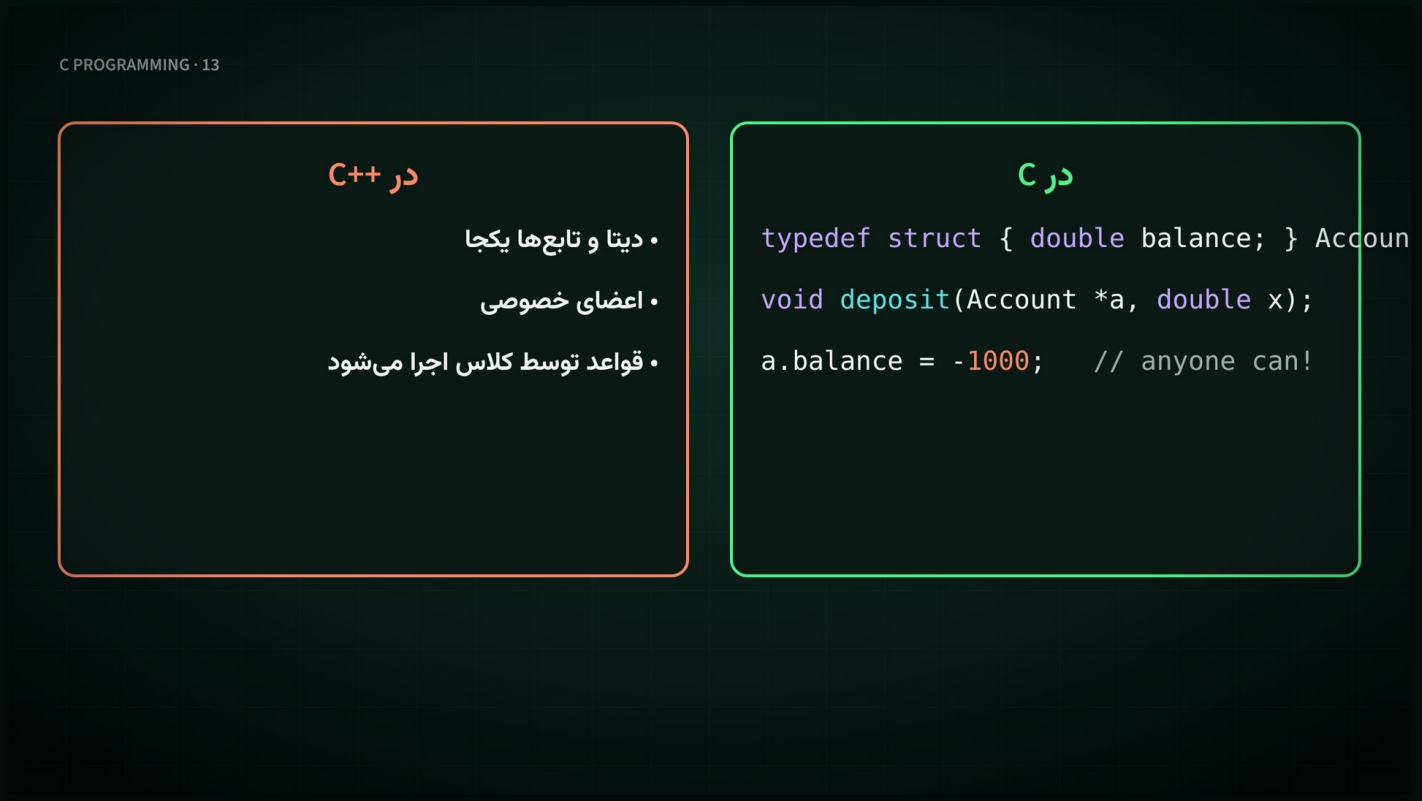
enum Light l = GREEN;
switch (l) {
    case RED:    printf("Stop\n"); break;
    case GREEN:  printf("Go\n");   break;
    case YELLOW: printf("Slow\n"); break;
}
```

۱۳.۰۶

مدل کردن دنیای واقعی

با struct، union، enum و typedef تقریباً هر چیزی را مدل کرده می‌توانید: یک حساب بانکی، ...

اما ببینید چه کم است: تابع‌هایی که روی یک struct کار می‌کنند از آن جدا زندگی می‌کنند، و هر ...



۱۳.۰۷

یک کورس رایگان از MyWebUniversity

برنامه‌نویسی C

درس ۱۴

لیست‌های پیوندی

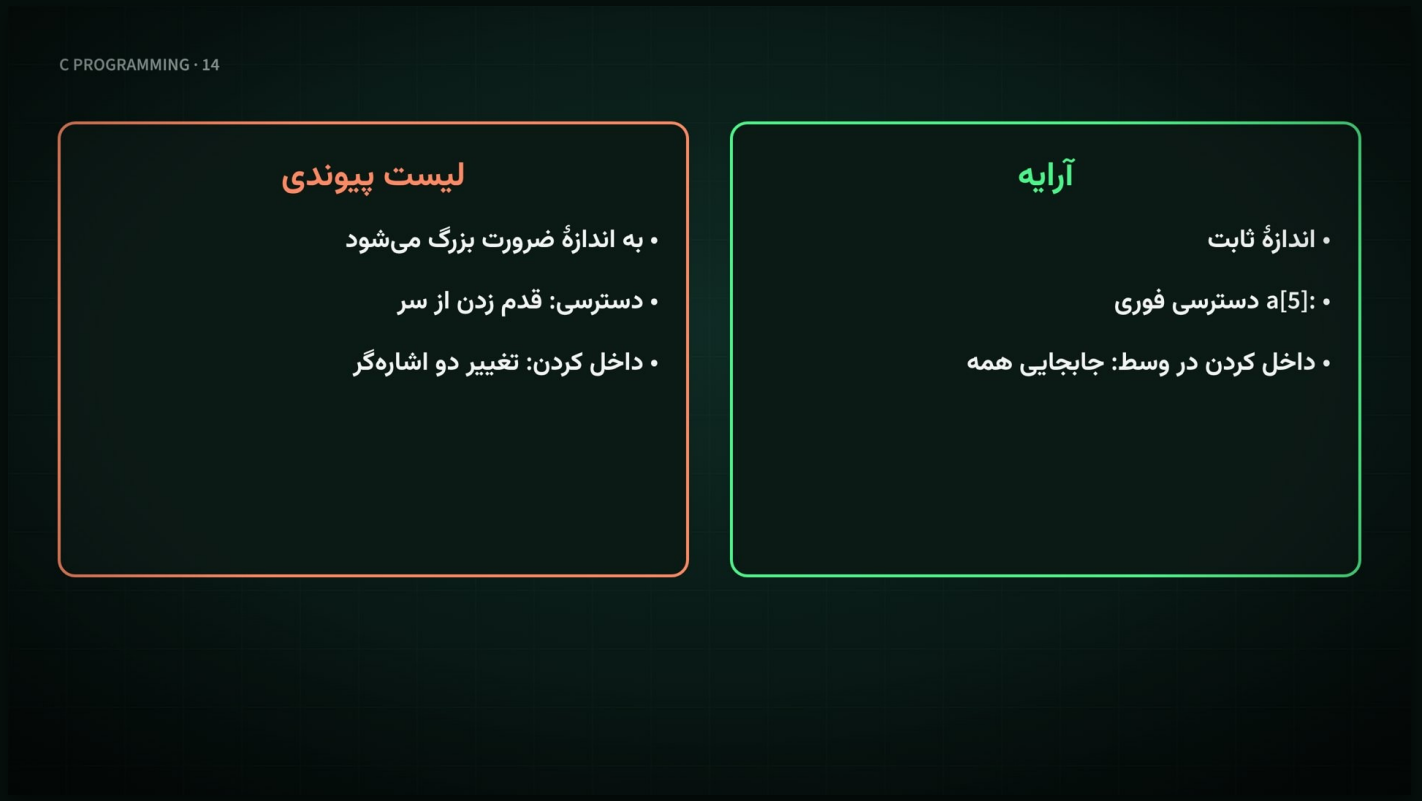
اولین ساختار دیتای پویای شما



چرا تنها آرایه نه؟

آرایه‌ها اندازهٔ ثابت دارند، و داخل کردن یک عنصر در وسط یعنی جابجا کردن همه چیز بعد از آن.

لیست پیوندی یک گره در هر بار بزرگ می‌شود، و داخل کردن یا حذف یک گره تنها چند اشاره‌گر را ...



۱۴.۰۱

گره

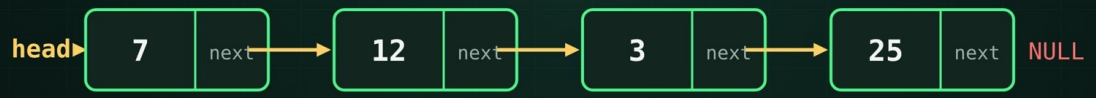
لیست پیوندی زنجیری از گره‌ها است. هر گره یک struct است که مقداری دیتا و اشاره‌گری به گره ...

گره اول سر نام دارد. اشاره‌گر بعدی گره آخر NULL است که پایان لیست را نشان می‌دهد.

C PROGRAMMING · 14

list.c

```c
typedef struct Node {
    int data;
    struct Node *next;
} Node;
```



۱۴.۰۲

اضافه کردن گره در اول

برای اضافه کردن یک گره در اول، یک گره نو را با malloc رزرو کنید، دیتا را ذخیره کنید، بعدی آن ...

چون تابع باید اشاره‌گر سر صدا کننده را تغییر دهد، اشاره‌گری به سر می‌گیرد، یعنی نود ستاره ...

C PROGRAMMING · 14

push.c

```c
int push(Node **head, int value) {
    Node *n = malloc(sizeof *n);
    if (n == NULL) return -1;
    n->data = value;
    n->next = *head;      // new node points to old head
    *head   = n;          // new node becomes the head
    return 0;
}

Node *list = NULL;
push(&list, 3); push(&list, 12); push(&list, 7);
```

۱۴.۰۳

قدم زدن در لیست

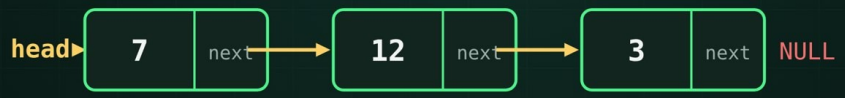
برای دیدن هر گره، از سر شروع کنید و اشاره‌گرهای بعدی را تا رسیدن به NULL دنبال کنید.

همین حلقه پایهٔ چاپ کردن، شمردن، جستجو و جمع کردن است. مهم‌ترین الگو برای لیست‌های پیوندی است.

C PROGRAMMING · 14

list.c

```c
for (Node *p = head; p != NULL; p = p->next) {
    printf("%d -> ", p->data);
}
printf("NULL\n");    // 7 -> 12 -> 3 -> NULL
```



۱۴.۰۴

حذف یک گره

برای حذف یک گره، گره قبلی را وادار کنید از روی آن بگذرد، بعد آن را آزاد کنید.

از اشاره‌گری به پیوندی که شاید تغییر کند استفاده کنید. آن وقت حذف سر و حذف گره وسط یک کود ...

C PROGRAMMING · 14

remove.c

```c
void remove_value(Node **link, int value) {
    while (*link != NULL) {
        if ((*link)->data == value) {
            Node *dead = *link;
            *link = dead->next;     // skip over it
            free(dead);             // give memory back
            return;
        }
        link = &(*link)->next;
    }
}
```

۱۴.۰۵

آزاد کردن تمام لیست

هر گره با malloc ساخته شده، پس هر گره باید آزاد شود. در لیست قدم بزنید، اما قبل از آزاد کردن ...

اگر اول آزاد کنید و بعد بعدی را بخوانید، از حافظهٔ آزادشده استفاده می‌کنید، همان باگ ...

C PROGRAMMING · 14

free_list.c

```c
void free_list(Node *head) {
    while (head != NULL) {
        Node *next = head->next;   // save first!
        free(head);
        head = next;
    }
}
```

۱۴.۰۶

خانوادهٔ ساختارهای پیوندی

دوطرفه — بعدی + قبلی

ستک — آخرین ورودی، اولین خروجی

صف — اولین ورودی، اولین خروجی



۱۴.۰۷

یک کورس رایگان از MyWebUniversity

برنامه‌نویسی C

درس ۱۵

پیش‌پردازنده و فایل‌های هِدر

#include، #define، ماکروها، محافظ‌ها، پروگرام چندفایلی و Makefile



#include واقعاً چه می‌کند

<file.h>> — هِدرهای سیستم

"file.h" — هِدرهای پروژه

gcc -E — دیدن نتیجه

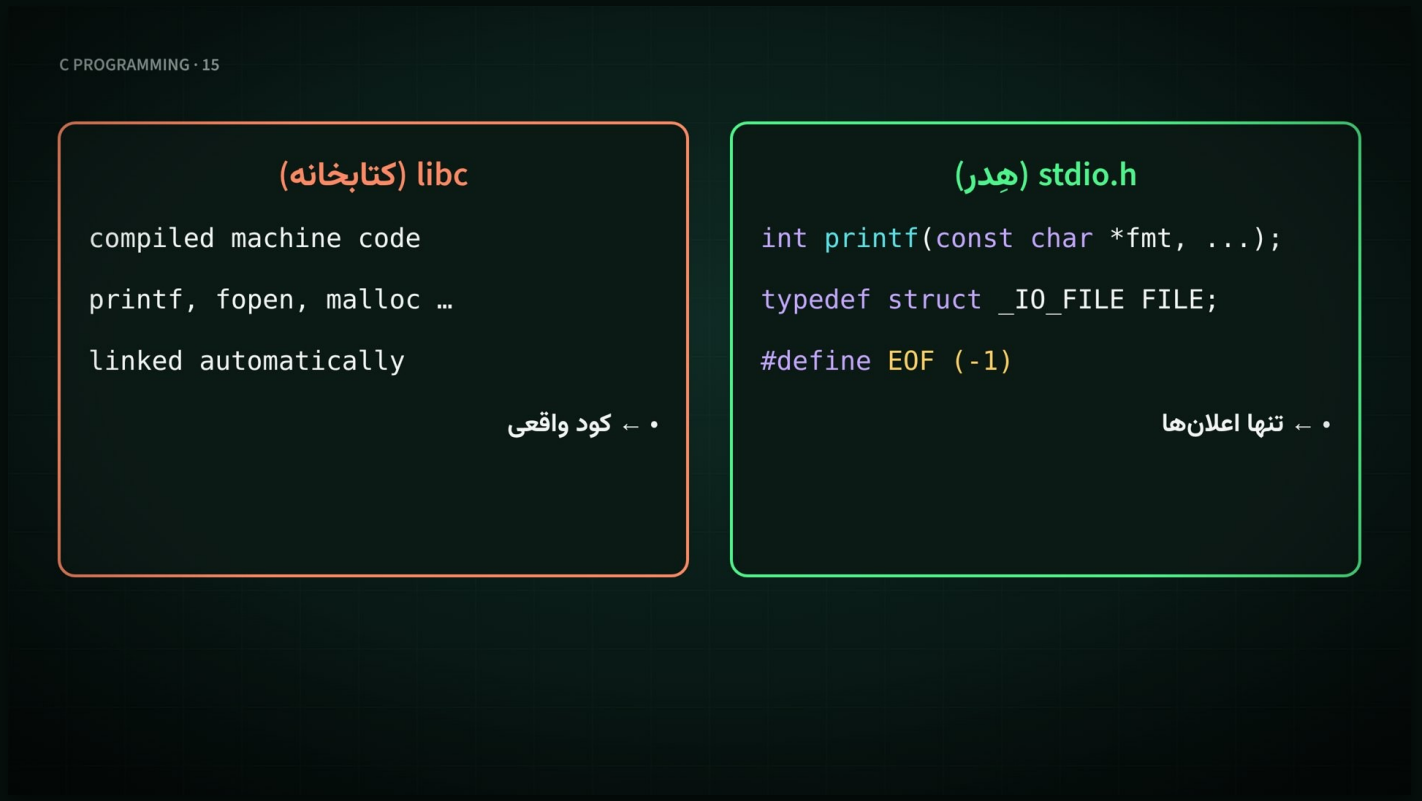


۱۵.۰۱

داخل stdio چیست؟

هِدری مثل stdio کود پرینت if را ندارد. اعلان‌ها را دارد: پروتوتایپ‌ها، تعریف ...

کود کمپایل‌شدهٔ واقعی در کتابخانهٔ C، یعنی libc، زندگی می‌کند که لینکر خودکار وصل ...



۱۵.۰۲

هِدرهای استاندارد رایج

هِدر	به شما می‌دهد
stdio.h	printf scanf fgets fopen fprintf FILE EOF
stdlib.h	malloc free atoi strtol qsort rand exit
string.h	strlen strcpy strcmp memcpy memset
math.h	sqrt pow sin fabs (link with -lm)
ctype.h · time.h	isdigit toupper · time clock strftime
stdint.h · stdbool.h	int32_t uint8_t · bool true false

این‌ها هِدرهایی است که بیشتر استفاده می‌کنید. stdio برای ورودی و خروجی، ...

math برای جذر، توان و سینوس؛ در Linux آن را با -lm لینک کنید. ctype برای حروف، ...

۱۵.۰۳

#define: ثابت‌ها و ماکروها

هش #define نامی می‌سازد که پیش‌پردازنده قبل از کمپایل آن را با متن جایگزین می‌کند. اکثراً ...

ماکروی پارامتردار مثل تابع معلوم می‌شود، اما فقط جایگزینی متن است، و باعث اتفاق‌های ...

حتی با پرانتز، ماکرو شاید آرگومان خود را دو بار حساب کند. برای بیشتر کارها، یک متغیر const ...

C PROGRAMMING · 15

macros.c

```c
#define MAX_STUDENTS 100
#define SQUARE_BAD(x)  x * x
#define SQUARE(x)      ((x) * (x))

SQUARE_BAD(2 + 3)   // 2 + 3 * 2 + 3 = 11  ✗
SQUARE(2 + 3)       // ((2+3) * (2+3)) = 25 ✓
SQUARE(i++)         // i++ runs twice!      ✗

static inline int square(int x) { return x * x; }   // safer
```

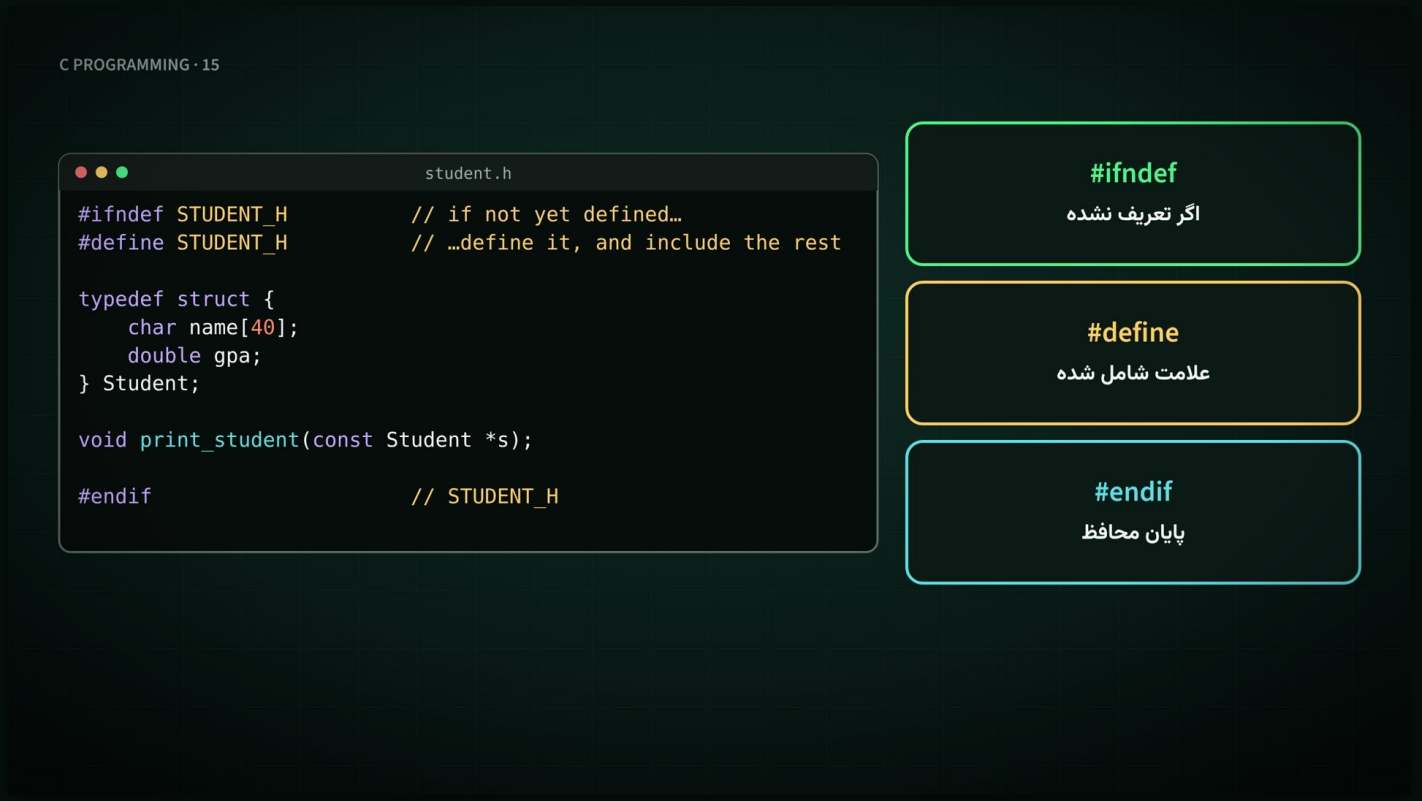
۱۵.۰۴

محافظ‌های #include: #ifndef

#ifndef — اگر تعریف نشده

#define — علامت شامل شده

#endif — پایان محافظ



۱۵.۰۵

کمپایل شرطی

#if، #ifdef و هش else اجازه می‌دهند برای حالت‌های مختلف کود مختلف کمپایل کنید، مثل ...

می‌توانید یک نام را از خط فرمان با D- تعریف کنید، پس همان کود منبع نسخهٔ دیباگ یا ...

debug.c

```c
#ifdef DEBUG
    #define LOG(msg) fprintf(stderr, "[debug] %s\n", msg)
#else
    #define LOG(msg) ((void)0)
#endif

#ifdef _WIN32
    const char *os = "Windows";
#else
    const char *os = "Linux / macOS";
#endif
```

```
$ gcc -DDEBUG -g app.c -o app
```

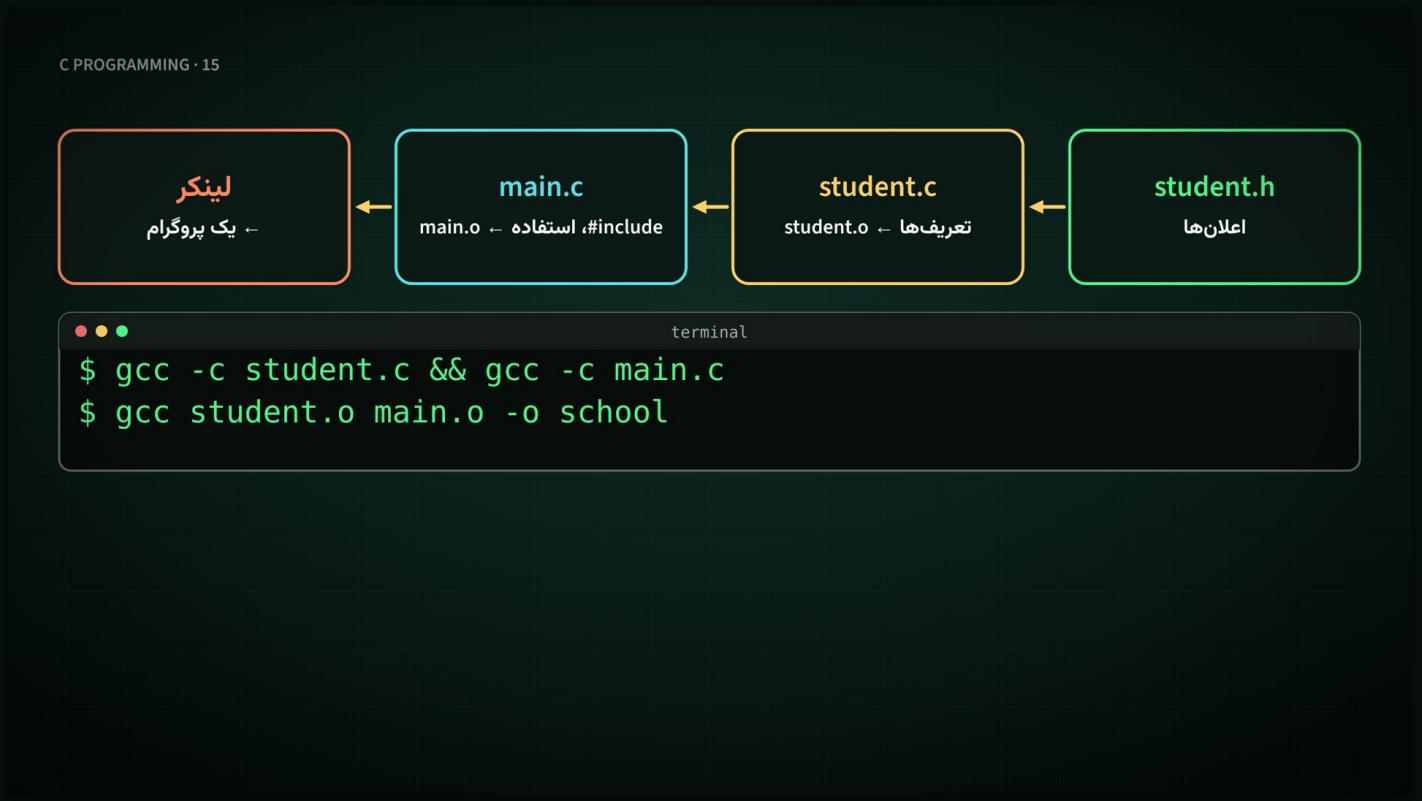
۱۵.۰۶

پروگرام‌های چندفایلی

student.h — اعلان‌ها

student.c — تعریف‌ها ← student.o

main.c — #include، استفاده ← main.o



۱۵.۰۷

Makefiles

تایپ کردن فرمان‌های کمپایل خسته‌کننده می‌شود.
میک یک Makefile را می‌خواند که هر هدف، ...

میک تنها چیزی را که تغییر کرده دوباره می‌سازد که
در پروژه‌های بزرگ وقت زیادی ذخیره می‌کند. ...

Makefile

```
CC     = gcc
CFLAGS = -std=c17 -Wall -Wextra -g

school: main.o student.o
 $(CC) main.o student.o -o school

%.o: %.c student.h
 $(CC) $(CFLAGS) -c $<

clean:
 rm -f *.o school

$ make        # builds only what changed
```

۱۵.۰۸

یک کورس رایگان از MyWebUniversity

برنامه‌نویسی C

درس ۱۶

کتابخانهٔ استاندارد و فایل‌ها

فایل‌ها، آرگومان‌های خط فرمان، qsort، اعداد تصادفی و خطاها

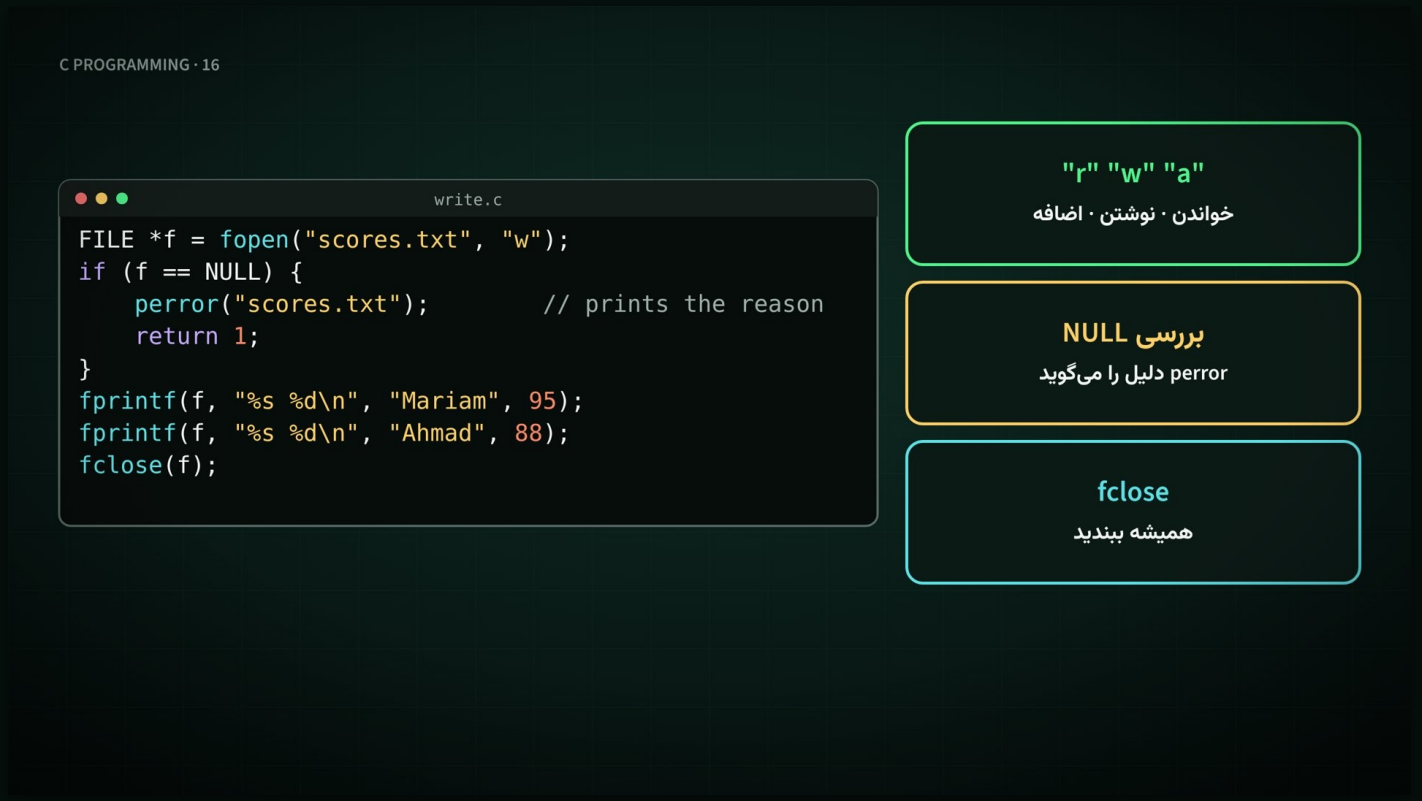


نوشتن یک فایل

"r" "w" "a" — خواندن · نوشتن · اضافه

بررسی NULL — perror دلیل را می‌گوید

fclose — همیشه ببندید



۱۶.۰۱

خواندن یک فایل

برای خواندن، با حالت r باز کنید و با fgets در یک حلقه یک سطر در هر بار بخوانید. اِف ...

برای بیرون کشیدن مقدارها از یک سطر، از sscanf استفاده کنید که مثل scanf است اما از ...

read.c

```c
FILE *f = fopen("scores.txt", "r");
if (!f) { perror("scores.txt"); return 1; }
char line[128], name[40];
int score;
while (fgets(line, sizeof line, f) != NULL) {
    if (sscanf(line, "%39s %d", name, &score) == 2) {
        printf("%-10s %3d\n", name, score);
    }
}
fclose(f);
```

۱۶.۰۲

فایل‌های باینری

فایل‌های متنی برای انسان‌ها آسان‌خوان هستند.
فایل‌های باینری بایت‌های خام دیتای شما را ...

با rb یا wb باز کنید، و از fwrite و fread برای ذخیره و بارگذاری تمام structها ...

C PROGRAMMING · 16

binary.c

```c
Student list[30];

FILE *out = fopen("class.dat", "wb");
fwrite(list, sizeof(Student), 30, out);
fclose(out);

FILE *in = fopen("class.dat", "rb");
size_t n = fread(list, sizeof(Student), 30, in);
fclose(in);
```

۱۶.۰۳

آرگومان‌های خط فرمان

main می‌تواند کلمه‌هایی را که بعد از نام پروگرام تایپ شده بگیرد. argc تعداد آرگومان‌ها ...

ابزارهایی مثل gcc و grep همین طور نام فایل‌ها و گزینه‌ها را می‌گیرند. همیشه قبل از ...

C PROGRAMMING · 16

greet.c

```c
int main(int argc, char *argv[]) {
    if (argc < 2) {
        fprintf(stderr, "usage: %s NAME\n", argv[0]);
        return 1;
    }
    printf("Hello, %s!\n", argv[1]);
    return 0;
}
```

```
$ ./greet Wahid
Hello, Wahid!
```

۱۶.۰۴

تابع‌های مفید stdlib

strtol متن را با اطمینان به عدد تبدیل می‌کند و می‌گوید عدد کجا تمام شد. atoi ساده‌تر ...

qsort هر آرایه را مرتب می‌کند؛ شما یک تابع مقایسه به آن می‌دهید، با استفاده از ...

exit پروگرام را از هر جا تمام می‌کند، و EXIT_SUCCESS و EXIT_FAILURE کودهای برگشتی روشن ...

C PROGRAMMING · 16

sort.c

```c
int cmp_int(const void *a, const void *b) {
    int x = *(const int *)a, y = *(const int *)b;
    return (x > y) - (x < y);
}

int v[] = {42, 7, 19, 3};
qsort(v, 4, sizeof v[0], cmp_int);     // 3 7 19 42

srand((unsigned)time(NULL));            // seed once
int dice = rand() % 6 + 1;              // 1..6
long n = strtol("123abc", &end, 10);   // 123, end → "abc"
```

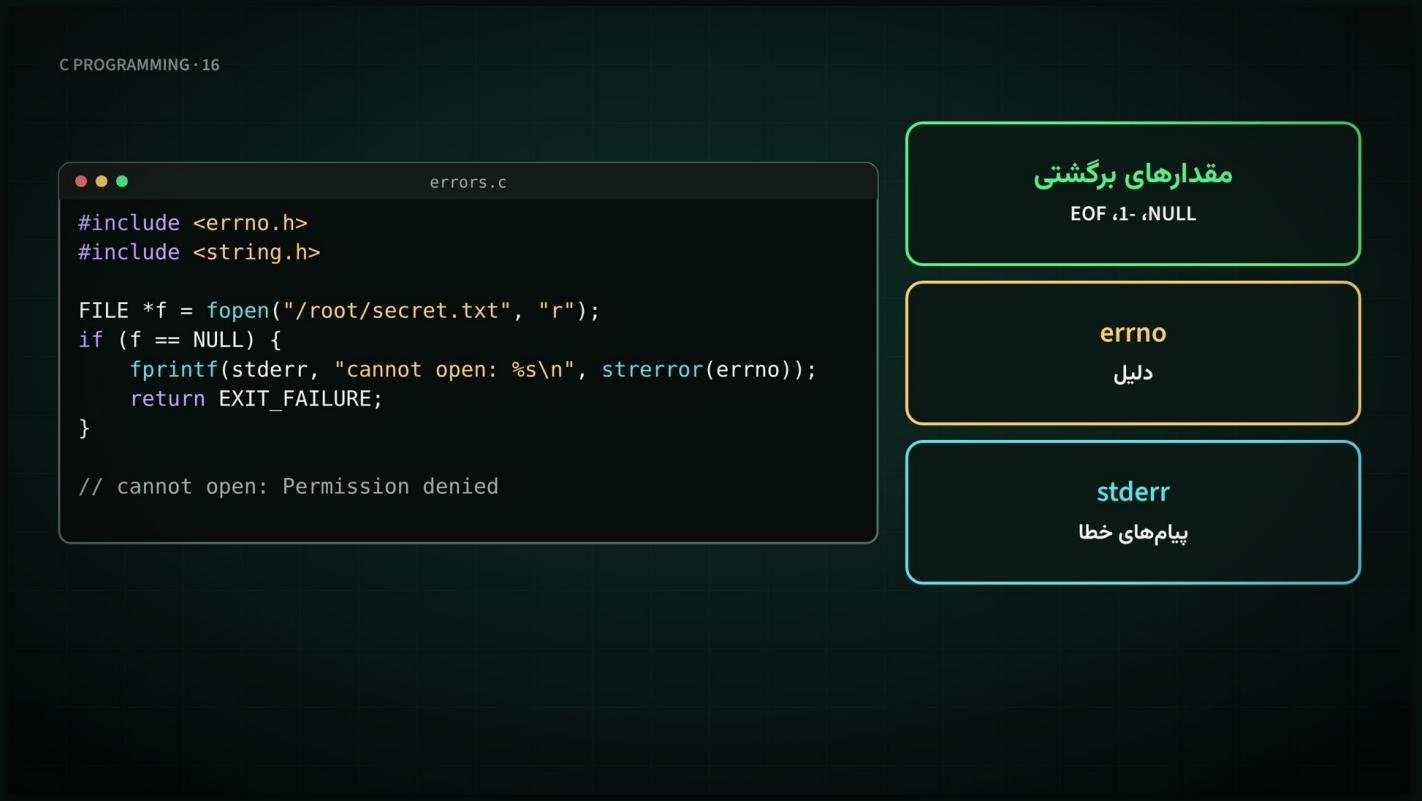
۱۶.۰۵

مدیریت خطاها

مقدارهای برگشتی — NULL، -1، EOF

errno — دلیل

stderr — پیام‌های خطا



۱۶.۰۶

یک کورس رایگان از MyWebUniversity

برنامه‌نویسی C

درس ۱۷

طراحی ماژولار، محدودیت‌های C، و راه به سوی C++

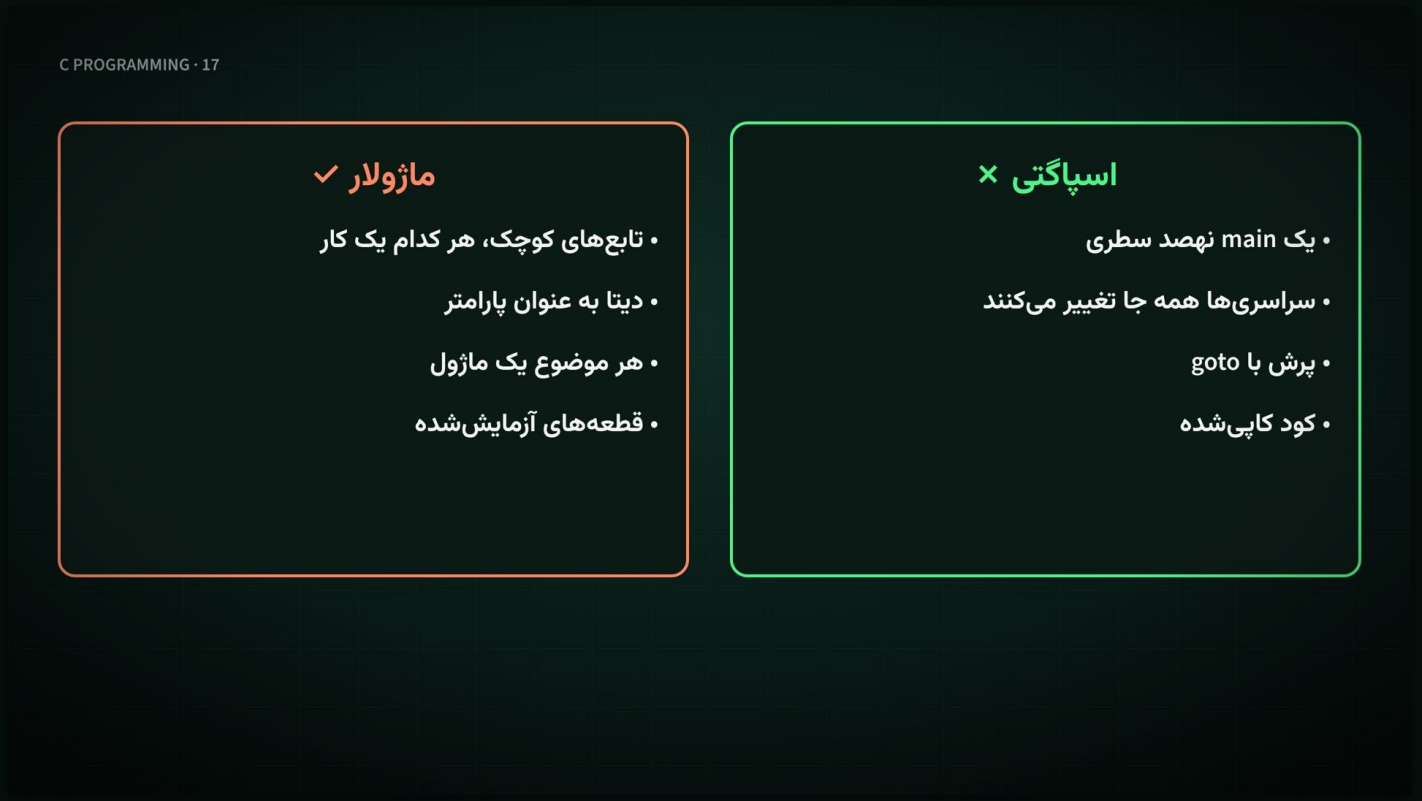
کود اسپاگتی در برابر ماژول‌های تمیز، و چرا C++ به دنیا آمد



کود اسپاگتی

کود اسپاگتی کودی است که همه چیز در آن به هم پیچیده: یک main بزرگ، متغیرهای سراسری که از همه ...

شاید اول کار کند، اما هر تغییر چیز دیگری را خراب می‌کند، و هیچ کس، حتی خودتان بعد از شش ...



۱۷.۰۱

ماژول‌ها در C

یک ماژول در C یک جفت فایل است: هِدری که رابط عمومی است، و یک فایل .c که پیاده‌سازی ...

تابع‌ها و متغیرهای static در ماژول خصوصی می‌مانند. استفاده‌کنندگان تنها هِدر را می‌بینند، ...

C PROGRAMMING · 17

stack.h · stack.c

```c
// stack.h — the public interface
typedef struct Stack Stack;          // opaque: details hidden
Stack *stack_new(void);
void   stack_push(Stack *s, int v);
int    stack_pop(Stack *s);
void   stack_free(Stack *s);

// stack.c — the private implementation
struct Stack { int *items; int top, cap; };
static void grow(Stack *s) { /* private helper */ }
```

۱۷.۰۲

کمبودهای C استاندارد

بدون کلاس — دیتا و تابع جدا

بدون پنهان‌سازی — هر کس struct را تغییر می‌دهد

بدون namespace — نام‌ها برخورد می‌کنند



۱۷.۰۳

متغیرها در برابر آبجکت‌ها

در C، متغیر فقط دیتا است. موجودی بانک یک double است که هر کودی می‌تواند آن را منفی هزار بسازد.

آبجکت دیتا را با تابع‌هایی که اجازهٔ تغییر آن را دارند ترکیب می‌کند و قواعد خود را حفظ ...

C: یک متغیر

```c
double balance = 100;

balance = -1000;   // nobody stops this
```

C++: یک آبجکت

```cpp
Account acc(100);

acc.deposit(-50);  // refused

acc.balance = 0;   // compile error
```

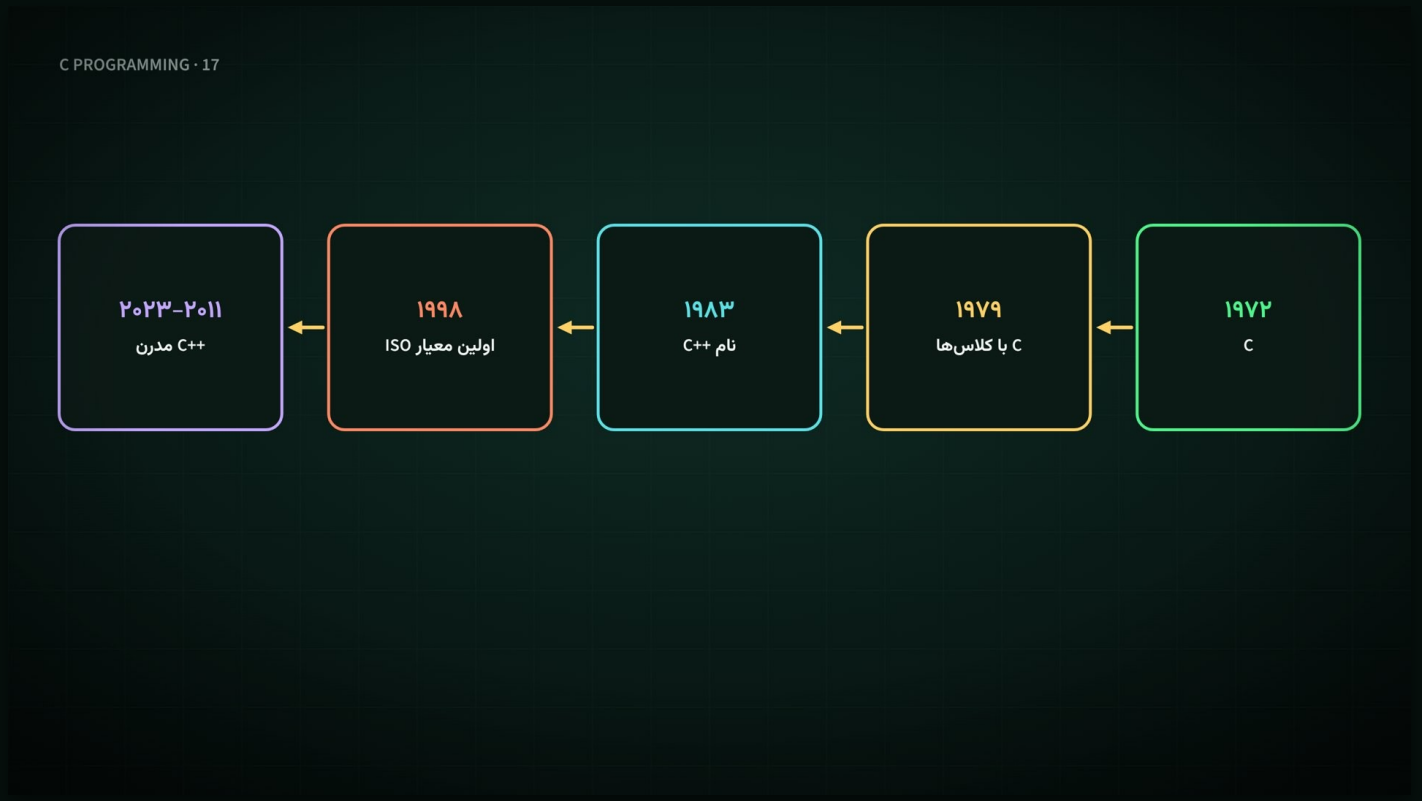
۱۷.۰۴

چرا C++ به دنیا آمد

C — ۱۹۷۲

C با کلاس‌ها — ۱۹۷۹

C++ نام — ۱۹۸۳



۱۷.۰۵

نگاهی به چهار ستون

کپسوله‌سازی — دیتا و تابع یکجا

تجرید — پنهان کردن جزئیات

وراثت — استفادهٔ دوباره و گسترش



۱۷.۰۶

تبریک، و قدم بعدی

مدیریت نمرات — struct · آرایه · فایل

شمارندهٔ کلمات — رشته · فایل · argv

بانک کوچک — لیست پیوندی · malloc



۱۷.۰۷

منابع و مطالعهٔ بیشتر (۱/۱)

1. Kernighan & Ritchie, The C Programming Language, 2nd ed., 1988
2. ISO/IEC 9899:2018 (C17) and 9899:2024 (C23) — the C standard
3. cppreference.com — C language and standard library reference
4. GCC manual — gcc.gnu.org/onlinedocs · GDB manual — sourceware.org/gdb
5. MyWebUniversity — free C and Linux courses

تشکر از توجه شما

MyWebUniversity.com

@MyWebUniversity-Dari همه‌را یاد بگیرید

آموزش رایگان تکنالوژی از سال ۲۰۰۰ — انگلیسی و دری