



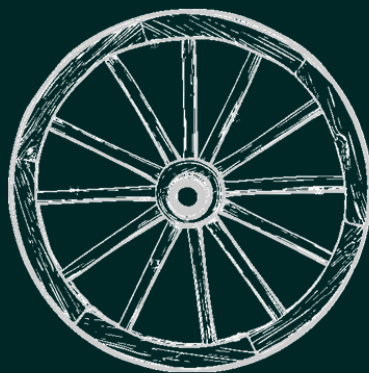
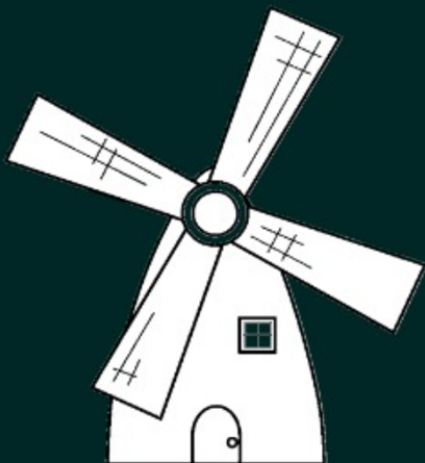
مقدمه‌ای کوتاه درباره کامپیوتر

تحلیل داده با پایتون

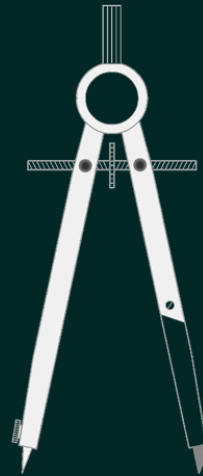
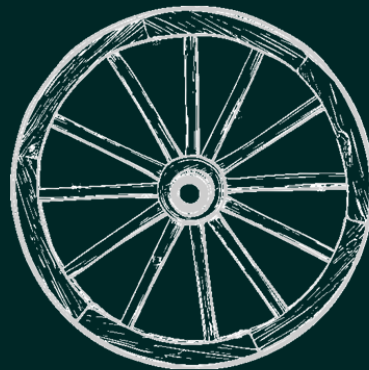
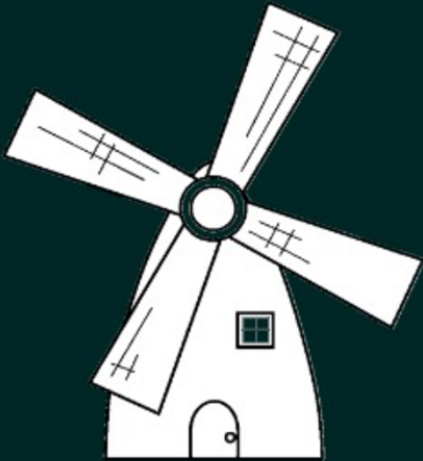
علیرضا کدیور

۱۴۰۵

انسان ابزارساز چه ابزارهایی ساخته است؟



انسان ابزارساز چه ابزارهایی ساخته است؟

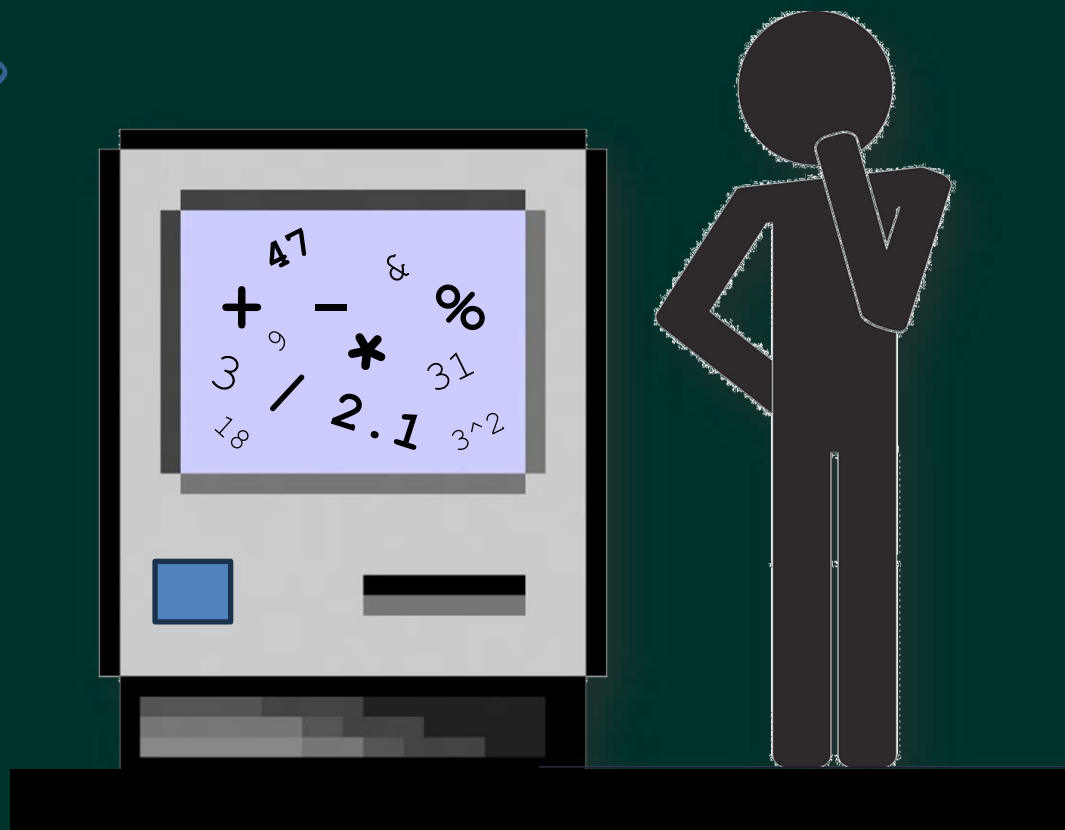


همه این ابزارها انسان را در کار دستی یا فعالیت بدنی یا به بیان کلی‌تر «فعالیت فیزیکی» کمک می‌کنند.

چرا انسان ابزاری به نام کامپیوتر را ابداع کرد؟

- برخلاف ابزارهای دیگر، کامپیوتر ساخته شد تا در نوع خاصی از فکر کردن «فعالیت ذهنی» به بشر کمک کند.

تفکر محاسباتی



تحويل مسئله به کامپیوتر برای کمک گرفتن از توان محاسباتی آن، با «الگوریتم» انجام می شود.

تفکر محاسباتی

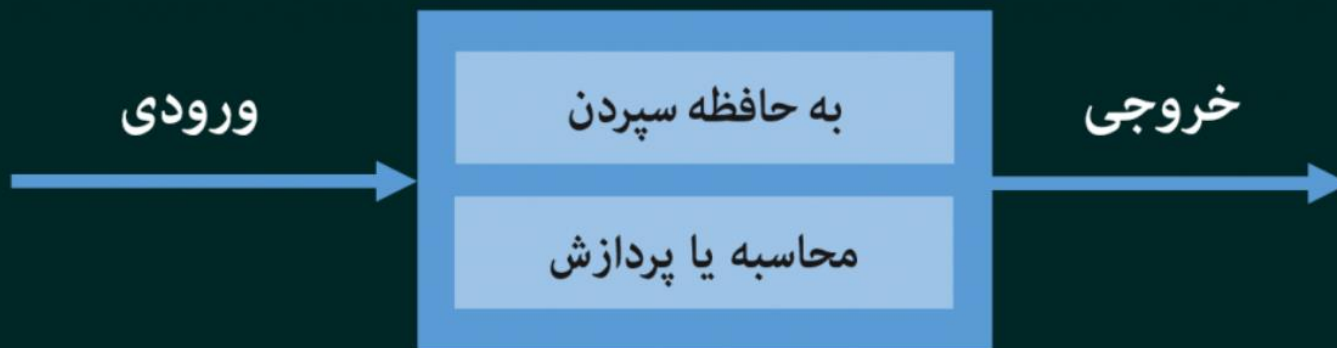
طرح مسئله به گونه ای که
اطلاعات مورد نیاز برای حل آن و پاسخ نهایی
کاملاً مشخص باشد.

راهکار ساده و گام به گام
برای حل مسئله
(الگوریتم)

ماشین حل مسئله (کامپیوتر)

کامپیوتر چیست؟

کامپیوتر در یک تعریف کلی ابزاری است برای حل مسائل محاسباتی و پردازشی که **اطلاعات** را به عنوان **ورودی** دریافت می‌کند، آن‌ها را در **حافظه** ثبت می‌کند، تحت مجموعه‌ای از دستورالعمل‌های مشخص و از پیش تعیین‌شده (**الگوریتم**) **پردازش** می‌کند و نتیجه نهایی را به عنوان **خروجی** ارائه می‌کند.



حافظه



چه تعداد نقطه زرد در این تصویر هست؟

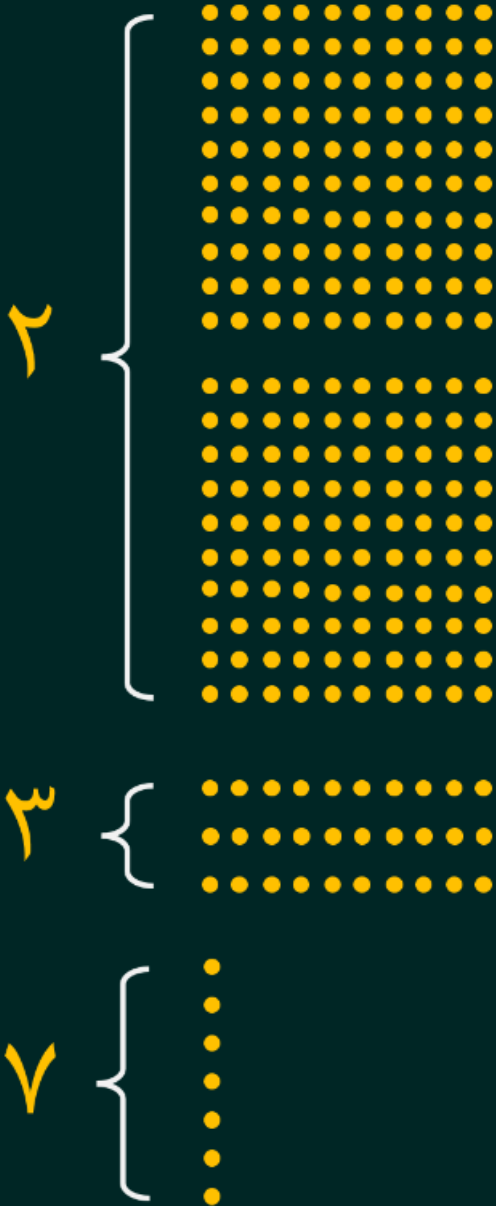
دستگاه دهدهی

$$237 = 10^2 \times 2 + 10^1 \times 3 + 1 \times 7$$

در دستگاه دهدهی 10 نماد برای ثبت وجود دارد

0 9 8 7 6 5 4 3 2 1

ما معمولاً از این دستگاه عددی استفاده می‌کنیم



دستگاه دودویی

برای شمارش تعداد مورد نظری می‌توانیم از یک دستگاه عددی با تعداد نماد کمتر استفاده کنیم. دستگاه عددی دودویی کمترین تعداد نماد را دارد.

$$\begin{aligned} 237 = & \mathbf{1} * 2^7 + \mathbf{1} * 2^6 + \mathbf{1} * 2^5 + \\ & \mathbf{0} * 2^4 + \mathbf{1} * 2^3 + \mathbf{1} * 2^2 + \\ & \mathbf{0} * 2^1 + \mathbf{1} * 2^0 \end{aligned}$$

$$(237)_{10} = (\mathbf{11101101})_2$$

دستگاه دودویی (استفاده از مبنای دو برای نمایش اعداد) کلید ثبت تعداد مورد نظر ما در کامپیوتر است و ابداع چیزی به نام **حافظه** برای ماشین است.

$$(237)_{10} = (11101101)_2$$

چرا؟!

11101101



تا اینجا توانستیم **اعداد** را با تبدیل به یک وضعیت دوتایی به خاطر بسپاریم و به واسطه پیاده کردن عملگرهای منطقی مانند و و یا و ... در قالب مدارهای الکتریکی روی آن محاسبات ریاضی هم انجام بدهیم.

دیگر انواع اطلاعات مانند متن، تصویر، صدا، دما، رطوبت و ... را چگونه به عنوان ورودی به کامپیوتر بدهیم؟

اگر بتوانیم انواع دیگر اطلاعات را هم به صورتی به عدد تبدیل کنیم می‌توانیم آن‌ها را هم وارد کامپیوتر کنیم. آیا این کار امکان‌پذیر است؟ چگونه؟

متن (Text)

قرارداد کردن مجموعه‌ای مشخص از اعداد برای نشان دادن حروف (Characters) کافیست تا بتوانیم اطلاعات متنی را هم در حافظه ثبت کنیم.

برای این کار توافقهایی برای کد کردن متن (حروف) در دنیا مورد توافق قرار گرفته که به آن‌ها استانداردهای کد کردن حرف (Character Encoding Standard) می‌گویند. مانند این موارد:

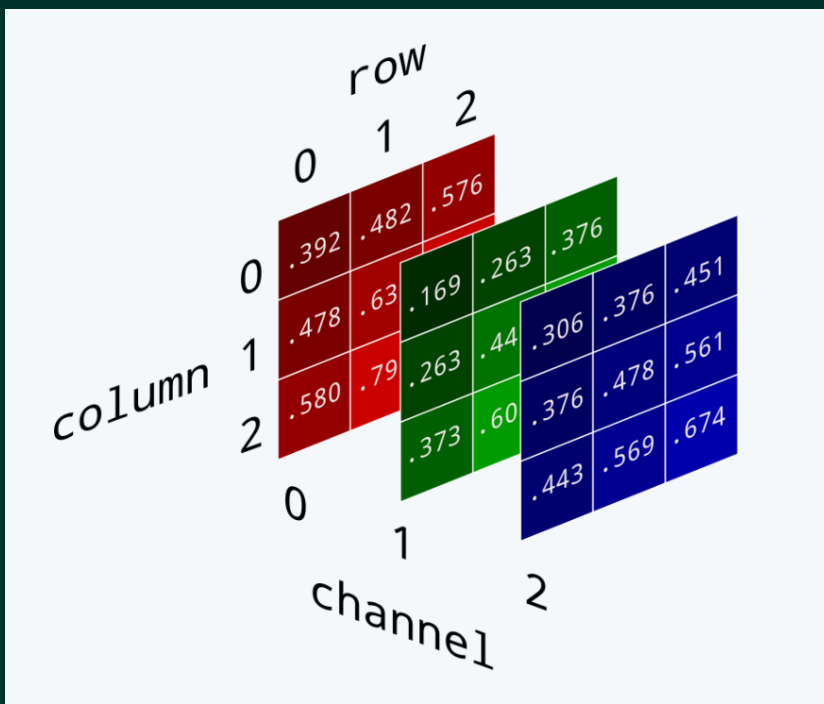
- ASCII (American Standard Code for Information Interchange)
- UTF-8
- UTF-16
- UTF-32

وقتی یک فایل متن با استاندارد (Character Encoding) اشتباهی خوانده و نمایش داده شود امکان استفاده از آن وجود ندارد. مثلاً این تصویر یک فایل است با استاندارد Windows-1256 خوانده شده در حالیکه اصل فایل قبلاً با استاندارد UTF-8 در کامپیوتر ذخیره شده که به همین دلیل امکان استفاده از آن وجود ندارد.

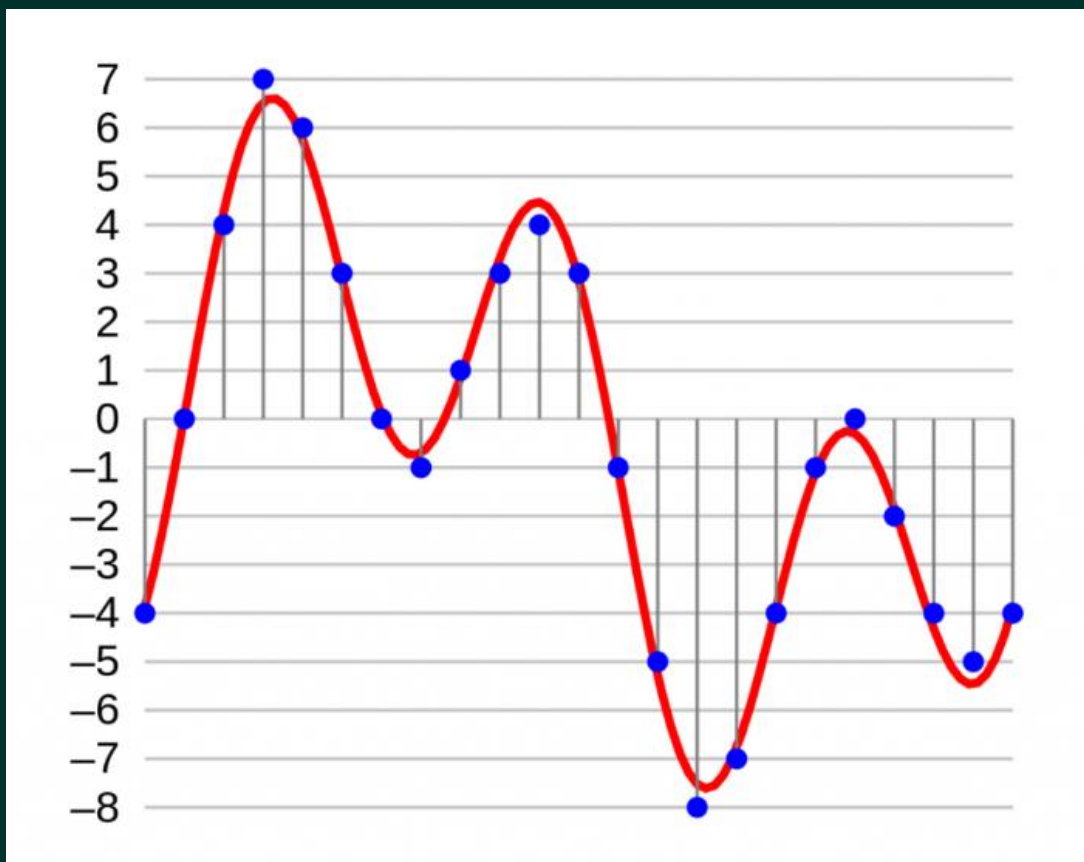
```
9 xÚµ[ms#·
10 b~¿B_<•SÖtoK.síL/N.u>6É>~Hn'kimm#iNAKíÊ>Ç×ETB À}³tŠ[wnæ
11
12 ~µ~<b3/~RE½!Úíªí~•ăFS³oŠûbÅ#nðÖc]ÖT«nÃh3'dpò)
13 EM9}¿Í7e¾ăpû!ÚTējİý/«öZ:)>²~;9tB<ĂúĂzÉbXîâă[-6 €ŸYKR'C?>
14 *`PV#iDC2
15 CBS00Ł&×kèðey_ÖlFDC4iEL-VTGSSENOTu`×ENO×òÍ,ÈPeBÊ`İDC4n
16 "ESCXi{ăSO}ETBSOLqİŠfü'no<²`û¾İiVTj¾-öESCö`«<t°cYBç»ò\N
17 ~CANZšeP FSÂCµ_ñ×>~ĒÍOBè=úăŸ...Gì-BELjÂSYNœ÷°œÇ^M±
18 ð÷Îœ÷;I7SOH|I2á+3üñĐ`CANô' àĒ-FSö²âĀgSUBöæ/ÛDC2' êĒ>šR,,
19 ²šİ.³DC3ETX0Z¥İiĀ²ZáXZNUSBG(ÈD!gXDC4i*
20 İiĀâVw°"DC2©ÇĀ^JÎ`X2YDD,Àš=-³SUBp+Öà[òMœ],3tSUB:~}ÿ.Îb=(
21 áüĀo'yªÓÚ;°-@óĀĒ-0USNULpMNUL<" :
22 '¬°ĎĀ--ò>_áÛşo,ÎöuYGSÑÇ,DC4cĀ>df,-cêøuòĐiŞNULSOH@á5wFSI
23 xÓ=<?ENOâIGó`„ĀûSYNNè²ª·%SUB© Û-İš
24 -NULªCANö-]òSTXXGENOúßwI[·x;ÿ%ENO
25 BELĀªl¾KÛ©]ò^fP»gí'DC1pt·zSYNİFē]USÍf»şdo>ÖÖŽ=ÉACK2ETX
26 Āçv^DC3±DC3SOHjELØ@ (EOTéà<BELİ©İĒSOHÉS±U-âœó÷LROI4FF°
27 !S~û &VTø†ŠªA`RSSTXâSUB,NULI[LV"÷¬¾İr°BELGSªw%ó0@"`NY@
28
```

تصویر

تعریف کوچک‌ترین واحد نمایش تصویر به نام پیکسل (Pixel)، تعریف موقعیت مکانی آن با دو عدد برای توصیف موقعیت افقی و موقعیت عمودی آن و سه عدد برای توصیف رنگ آن: **قرمز** **سبز** **آبی** برای ساختن رنگ‌های دیگر از ترکیب این سه رنگ (به این دستگاه ثبت رنگ‌ها اصطلاحاً **RGB** می‌گویند).



صوت

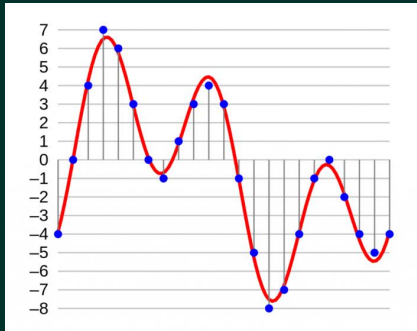


در این شکل، صوت به شکل یک موج با رنگ قرمز نمایش داده شده و در یک فاصله 4 بیتی برای ثبت 16 وضعیت ممکن (از -8 تا 7) از حالت پیوسته (آنالوگ) به گسسته (دیجیتال) تبدیل شده و با رنگ آبی نمایش داده شده است.

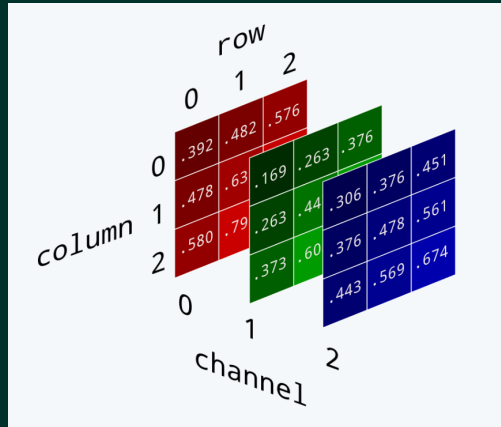
اطلاعات

010010101110

صدا



تصویر



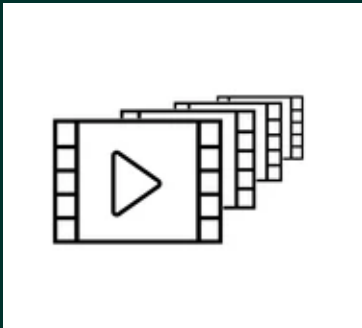
متن

```
*14020703.py - Notepad
File Edit Format View Help
# factorial function iterative
def f(n):
    N = 1
    for i in range(1,n+1):
        N = N * i
    return N
f(5)
# factorial function recursive
def f(n):
    if n == 1:
        return 1
```

عدد

23462

ویدئو



دما رطوبت فشار ...

اطلاعات

تعریف؟

اطلاعات

تعریف؟

هر تفاوتی که تفاوت آفرین باشد.

اطلاعات

تعریف؟

هر تفاوتی که تفاوت آفرین باشد.

0 $\longleftrightarrow$ 1

اطلاعات

binary digit

تعریف؟

هر تفاوتی که تفاوت آفرین باشد.



bit

0 ↔ 1

اطلاعات

binary digit

تعریف؟

هر تفاوتی که تفاوت آفرین باشد.



bit

0 ↔ 1



بیت: کوچک ترین واحد حافظه

اطلاعات

binary digit

تعریف؟

هر تفاوتی که تفاوت آفرین باشد.

bit

0 ↔ 1



بیت: کوچک ترین واحد حافظه

اطلاعات

binary digit

تعریف؟

هر تفاوتی که تفاوت آفرین باشد.



bit

0 ↔ 1



بیت: کوچک ترین واحد حافظه

اطلاعات

binary digit

تعریف؟

هر تفاوتی که تفاوت آفرین باشد.

bit

0 ↔ 1



بیت: کوچک‌ترین واحد حافظه

اطلاعات

binary digit

تعریف؟

هر تفاوتی که تفاوت آفرین باشد.

bit

0 ↔ 1



بیت: کوچک ترین واحد حافظه

اطلاعات

binary digit

تعریف؟

هر تفاوتی که تفاوت آفرین باشد.

bit

0 ↔ 1



بیت: کوچک ترین واحد حافظه

اطلاعات

binary digit

تعریف؟

هر تفاوتی که تفاوت آفرین باشد.



bit

0 ↔ 1



بیت: کوچکترین واحد حافظه

اطلاعات

binary digit

تعریف؟

هر تفاوتی که تفاوت آفرین باشد.

bit

0 ↔ 1



بیت: کوچک ترین واحد حافظه

اطلاعات

binary digit

تعریف؟

هر تفاوتی که تفاوت آفرین باشد.



bit

0 ↔ 1



بیت: کوچکترین واحد حافظه

اطلاعات

binary digit

تعریف؟

هر تفاوتی که تفاوت آفرین باشد.

bit

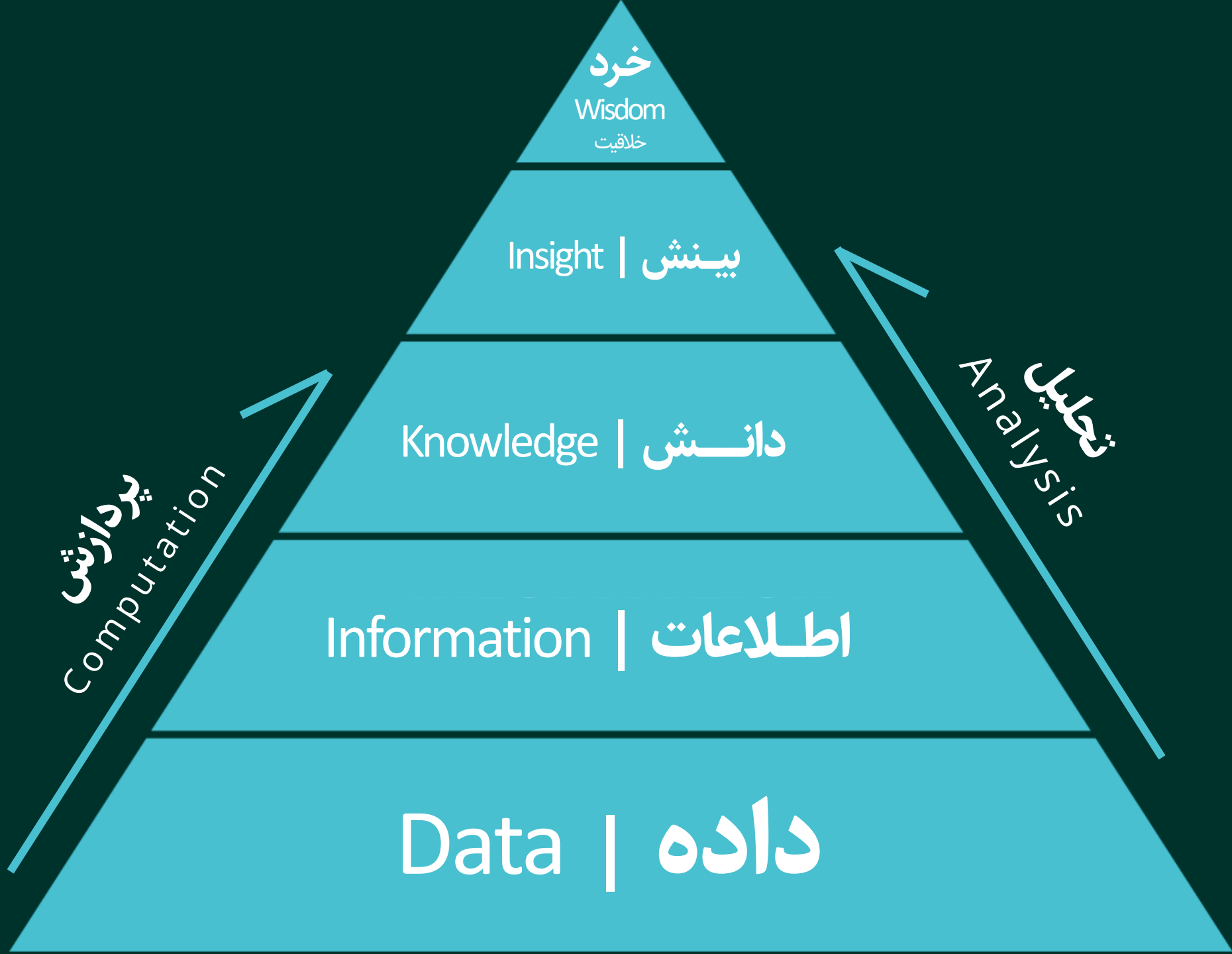
0 ↔ 1



بیت: کوچک ترین واحد حافظه

واحدهای اندازه‌گیری اطلاعات

واحد	تعداد بایت معادل
بیت (b)	1 / 8
بایت (B)	1
کیلوبایت (KB)	$2^{10} = 1,024$
مگابایت (MB)	$2^{20} = 1,048,576$
گیگابایت (GB)	$2^{30} = 1,073,741,824$
ترابایت (TB)	$2^{40} = 1,099,511,627,776$
پتابایت (PB)	$2^{50} = 1,125,899,906,842,624$



هر یک از علوم موضوعی مشخص برای مطالعه دارند

جامعه‌شناسی

شیمی

فیزیک

روان‌شناسی

آمار

زیست‌شناسی

هر یک از علوم موضوعی مشخص برای مطالعه دارند

جامعه‌شناسی

شیمی

فیزیک

روان‌شناسی

آمار

زیست‌شناسی

علوم کامپیوتر

?

موضوع اصلی:

اطلاعات
Information

پردازش
Computation

انفورماتیک
Informatics

علوم کامپیوتر
Computer Science

واژه سازی:

پردازش

تا اینجا دیدیم داده‌های متنی چگونه با تبدیل به صفر و یک در کامپیوتر ذخیره می‌شوند. برای اینکه ببینیم محاسبه و پردازش چگونه در کامپیوتر پیاده‌سازی می‌شود لازم است اول گریز کوتاهی به **منطق** بزنیم.

گریزی به منطق

منظور از گزاره منطقی چیست؟

چگونه گزاره‌های منطقی را با یکدیگر تلفیق می‌کنیم؟

به دو نمونه زیر توجه کنید.

نمونه اول: مجرم یا متهم؟! | عطف منطقی (و)

فرض کنید از مجموعه گزارش‌هایی که پلیس در ارتباط با یک صحنه جرم جمع‌آوری کرده گزاره‌های زیر استخراج شده است:

۱. مجرم یک مرد است.

۲. مجرم حدود ۴۵ تا ۵۵ سال سن دارد.

۳. مجرم قد بلند است.

در اداره آگاهی شخصی به عنوان متهم دستگیر شده و گفته شده

۴. این متهم همان مجرمی است که دنبالش بودیم.

اما گزاره زیر درباره متهم صادق است:

۵. متهم مردی قدبلند و ۲۶ ساله است.

با توجه به این موارد به نظر شما گزاره شماره ۴ می‌تواند درست باشد؟ چرا؟

نمونه سوم: جویای کار | فصل منطقی (یا)

فرض کنید شما مسئولیت پیدا کردن مددکار اجتماعی برای یک بنیاد خیریه را پذیرفته‌اید و باید فردی را پیدا کنید که دست کم یکی از شرایط زیر را داشته باشد.

۱. تجربه مددکاری اجتماعی در یک سازمان مردم‌نهاد را داشته باشد.

۲. در شاخه‌های مرتبط علوم اجتماعی مانند مددکاری اجتماعی، روانشناسی، جامعه‌شناسی، انسان‌شناسی، علوم تربیتی و ... تحصیل کرده باشد.

۳. معرفّی با سابقه و معتبر در فعالیتهای اجتماعی گواهی کند که متقاضی انگیزه و توان یادگیری و رشد در این کار را دارد.

لیلا رزومه‌اش را برای شما فرستاده و ضمن اظهار علاقه درخواست کرده این مسئولیت به او سپرده شود. او یک معلم است، پنج سال در همین کار سابقه دارد و در رشته تعلیم و تربیت تحصیل کرده است. با توجه به این اطلاعات به نظر شما گزاره ۴ درست است یا نادرست؟

۴. لیلا شرایط اولیه مددکار اجتماعی مورد نظر را دارد.

پایه‌ای‌ترین محاسبات

شاید بتوان گفت از محاسبات منطقی مانند «و» و «یا» در مثال‌های قبل، به عنوان ابتدایی‌ترین شکل پردازش نام برد که امکانات محاسباتی دیگر با استفاده از آنها ساخته می‌شوند.

«و»

A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

فرض کنید **A** یک گزاره منطقی است که می‌توانید صادق یا کاذب باشد، **B** هم همینطور. هر وقت یک گزاره کاذب بود بجای آن **0** و هر وقت صادق بود به جای آن **1** می‌گذاریم.

اگر بخواهیم عملگری مانند «و» یا «**AND**» برای آن تعریف کنیم این عملگر باید در حالت‌های مختلف چگونه کار کند؟ یعنی به ازای چه ورودی‌هایی چه خروجی‌هایی دریافت کند؟

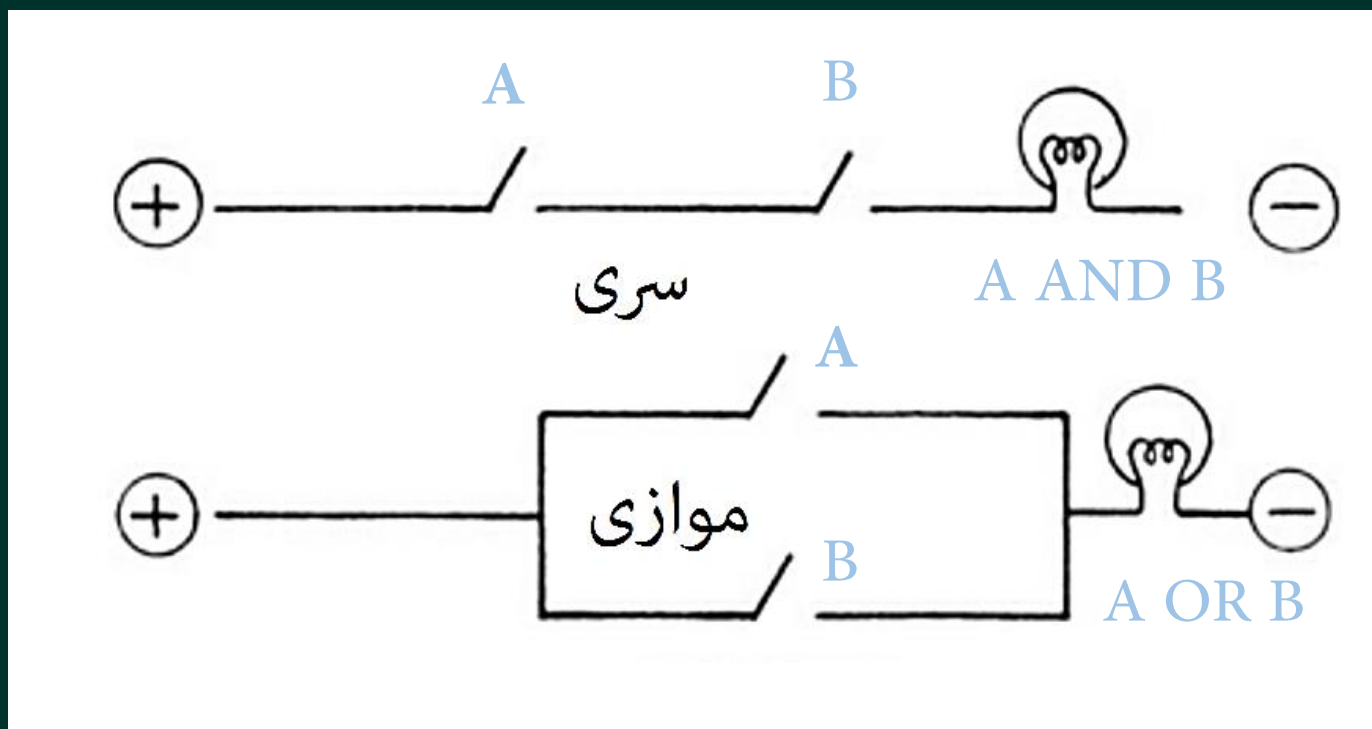
«یا»

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

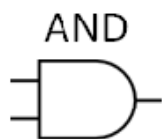
اگر بخواهیم عملگری مانند «یا» یا «OR» برای آن تعریف کنیم این عملگر باید در حالت‌های مختلف چگونه کار کند؟ یعنی به ازای چه ورودی‌هایی چه خروجی‌هایی دریافت کند؟

چگونه می‌توانیم یک عملگرهای **AND** و **OR**
را به صورت یک مدار الکتریکی درست کنیم؟

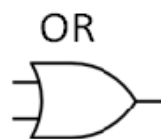
چگونه می‌توانیم یک عملگرهای **AND** و **OR** را به صورت یک مدار الکتریکی درست کنیم؟



در دنیای دیجیتال این کار را با استفاده از ادوات الکترونیکی مانند ترانزیستورها انجام می‌دهند و به آن **دروازه منطقی (Logic Gate)** می‌گویند و با قرار دادن آن در مدار الکتریکی یک یا دو جریان ورودی را دریافت و به جریان خروجی مورد نظر تبدیل می‌کنیم. برای نمایش دروازه‌های منطقی از اشکال استاندارد استفاده می‌شود.



INPUT		OUTPUT
A	B	
0	0	0
1	0	0
0	1	0
1	1	1



INPUT		OUTPUT
A	B	
0	0	0
1	0	1
0	1	1
1	1	1



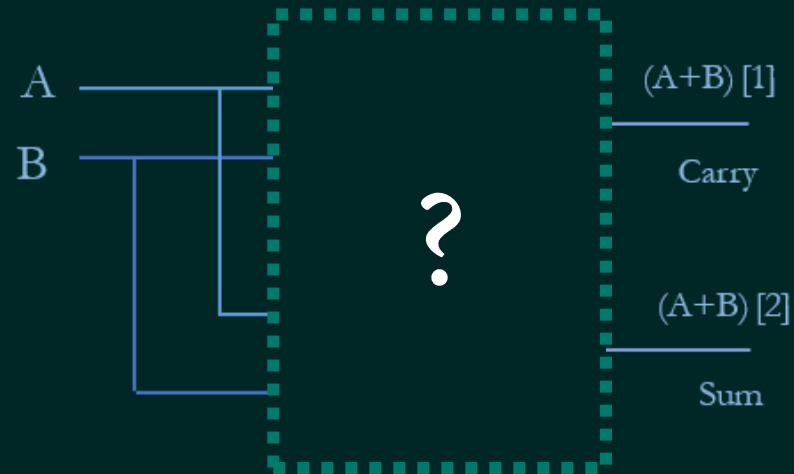
INPUT		OUTPUT
A	B	
0	0	0
1	0	1
0	1	1
1	1	0



INPUT	OUTPUT
A	
0	1
1	0

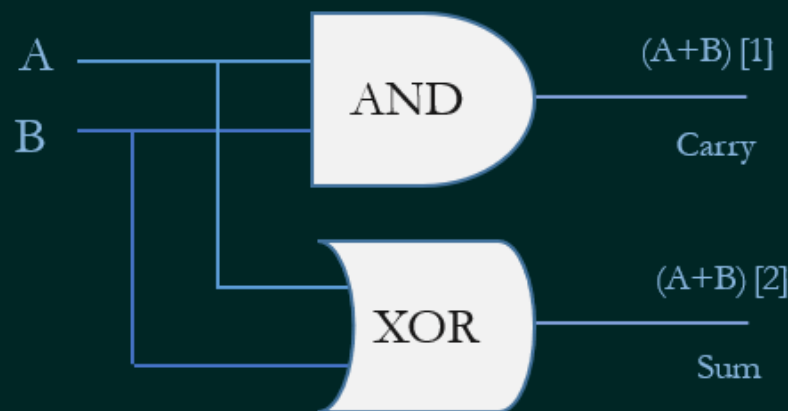
با استفاده از دروازه‌های منطقی می‌توان محاسبات پیچیده‌تر را به صورت فیزیکی به وجود آورد. مثلاً شکل زیر نمونه‌ای از جمع 2 بیتی است.

A	B	A + B
0	0	00
0	1	01
1	0	01
1	1	10



با استفاده از دروازه‌های منطقی می‌توان محاسبات پیچیده‌تر را به صورت فیزیکی به وجود آورد. مثلاً شکل زیر نمونه‌ای از جمع 2 بیتی است.

A	B	A + B
0	0	00
0	1	01
1	0	01
1	1	10



با استفاده از دروازه‌های منطقی می‌توان محاسبات پیچیده‌تر را به صورت فیزیکی به وجود آورد. مثلاً شکل زیر نمونه‌ای از جمع 2 بیتی است.

A	B	A + B
0	0	00
0	1	01
1	0	01
1	1	10





می‌توانیم با کنار هم
قرار دادن جمع‌های
دوبیتی که ساختیم
در یک مدار بزرگ‌تر
جمع ۸ بیتی
بسازیم.

جمع ۸ بیتی

می‌توانیم با کنار هم
قرار دادن جمع‌های
دوبیتی که ساختیم
در یک مدار بزرگ‌تر
جمع ۸ بیتی
بسازیم.

41 19

60

1 1

=

0

0 1

=

0

0 0

=

1

1 0

=

1

0 1

=

1

1 0

=

1

0 0

=

0

0 0

=

0

0

جمع
۸ بیتی

می‌توانیم با کنار هم
قرار دادن جمع‌های
دوبیتی که ساختیم
در یک مدار بزرگ‌تر
جمع ۸ بیتی
بسازیم.

$$(41)_{10} + (19)_{10} = (60)_{10}$$
$$(00101001)_2 + (00010011)_2 =$$
$$(000111100)_2$$

41 19

60

1 1

=

0

0 1

=

0

0 0

=

1

1 0

=

1

0 1

=

1

1 0

=

1

0 0

=

0

0 0

=

0

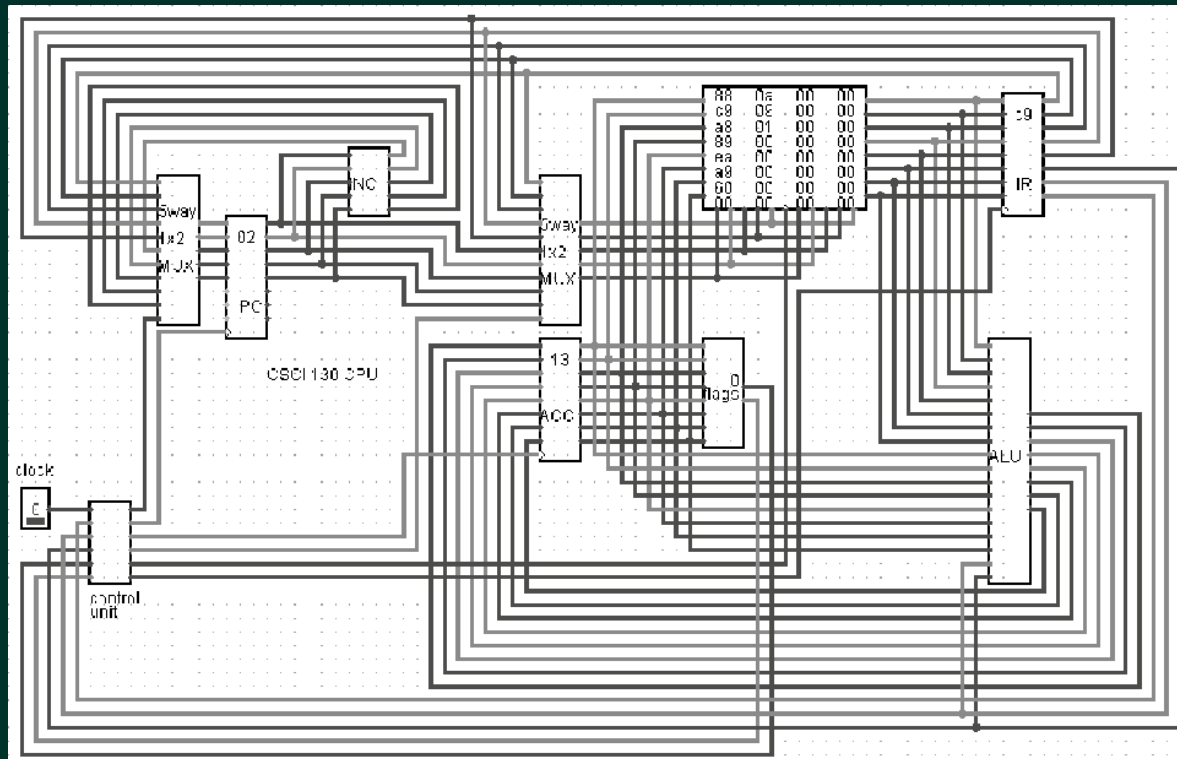
0

جمع
۸ بیتی

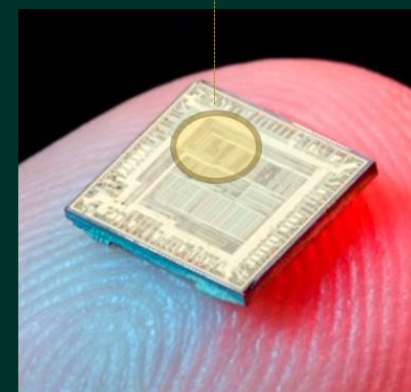
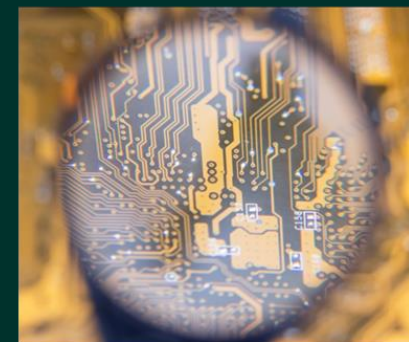
می‌توانیم با کنار هم
قرار دادن جمع‌های
دوبیتی که ساختیم
در یک مدار بزرگ‌تر
جمع ۸ بیتی
بسازیم.

$$(41)_{10} + (19)_{10} = (60)_{10}$$
$$(00101001)_2 + (00010011)_2 =$$
$$(000111100)_2$$

سازه‌های محاسباتی پیچیده‌تر همه مجموعه‌ای از همین دروازه‌های منطقی ساده هستند که در مدارهای منطقی بزرگ‌تر کنار یکدیگر قرار می‌گیرند و عملیات‌های پیچیده را امکان‌پذیر می‌کنند.



مدارهایی که برای استفاده در محاسبه‌گرها ساخته می‌شدند در طول زمان کوچک‌تر و کوچک‌تر شدند.



Our World
in Data

Transistor count

10,000,000,000

5,000,000,000

1.000.000.000

500.000.000

100,000,000

50,000,000

10.000.000

5,000,000

1,000,000

500,000

100,000

50,000

10,000

5,000

1,000



هر دو سال قدرت پردازشگرها
(تعداد ترانزیستورها در یک
تراشه با مساحت ثابت) تقریبا
دو برابر می‌شود.

Year in which the microchip was first introduced

Licensed under [CC-BY](#) by the authors Hannah Ritchie and Max Roser.

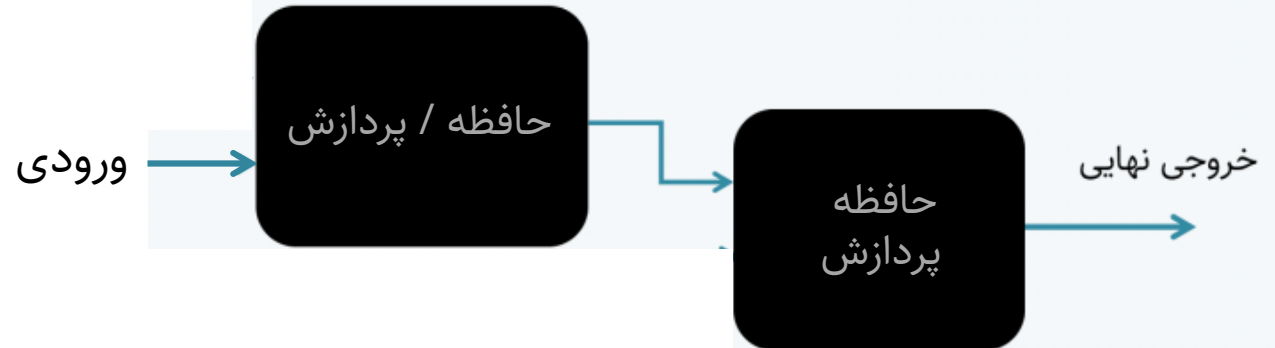
اکنون دریافتیم اطلاعات یا داده‌های خام چگونه در ابتدایی‌ترین شکل خود در **حافظه** کامپیوتر ذخیره و **پردازش** می‌شوند و با در کنار هم قرار گرفتن همین اجزای ابتدایی چگونه اشکال پیچیده‌تر اطلاعات در حافظه ثبت می‌شود یا اشکال پیچیده‌تر پردازش ممکن می‌شود.

در واقع، کامپیوتر **ساختار سلسله‌مراتبی** دارد. اجزای پیچیده‌تر، از اجزای ابتدایی‌تر ساخته می‌شوند و خود دوباره در ساخت اجزای پیچیده‌تر به کار گرفته می‌شوند. **انتزاع عملکرد** مفهومی بنیادین در پیداش و پیشرفت کامپیوتر است.

انتزاع عملکرد و ساختار سلسله مراتبی



انتزاع عملکرد و ساختار سلسله مراتبی



انتزاع عملکرد و ساختار سلسله مراتبی

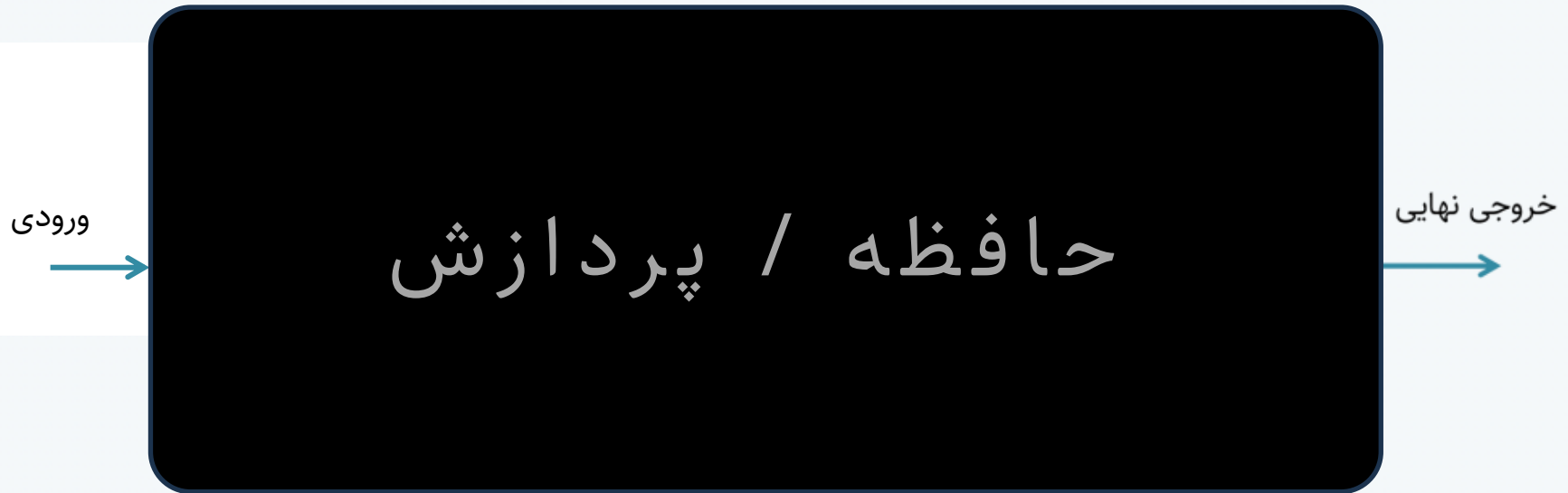


انتزاع عملکرد و ساختار سلسله مراتبی



انتزاع و ساختار سلسله مراتبی





نتیجه : به وجود آمدن سیستم‌های پیچیده، پیدایش موجودیت‌های جدید و ...

انتزاع و ساختار سلسله مراتبی



رویکرد ماژولار (پیمانه‌ای)
در طراحی سیستم

نتیجه : عیب‌یابی کارآمد، تقسیم کار و توسعه

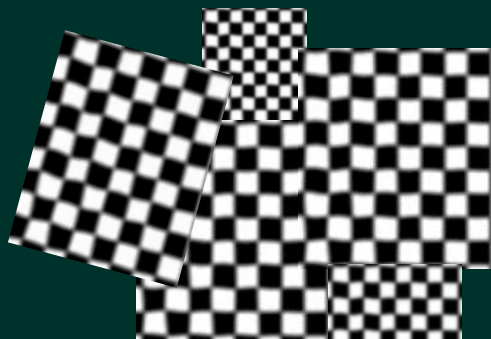
انتزاع عملکرد و ساختار سلسله مراتبی

فرض کنید می‌خواهیم با استفاده از مربع‌های زیر نقاشی بکشیم ...

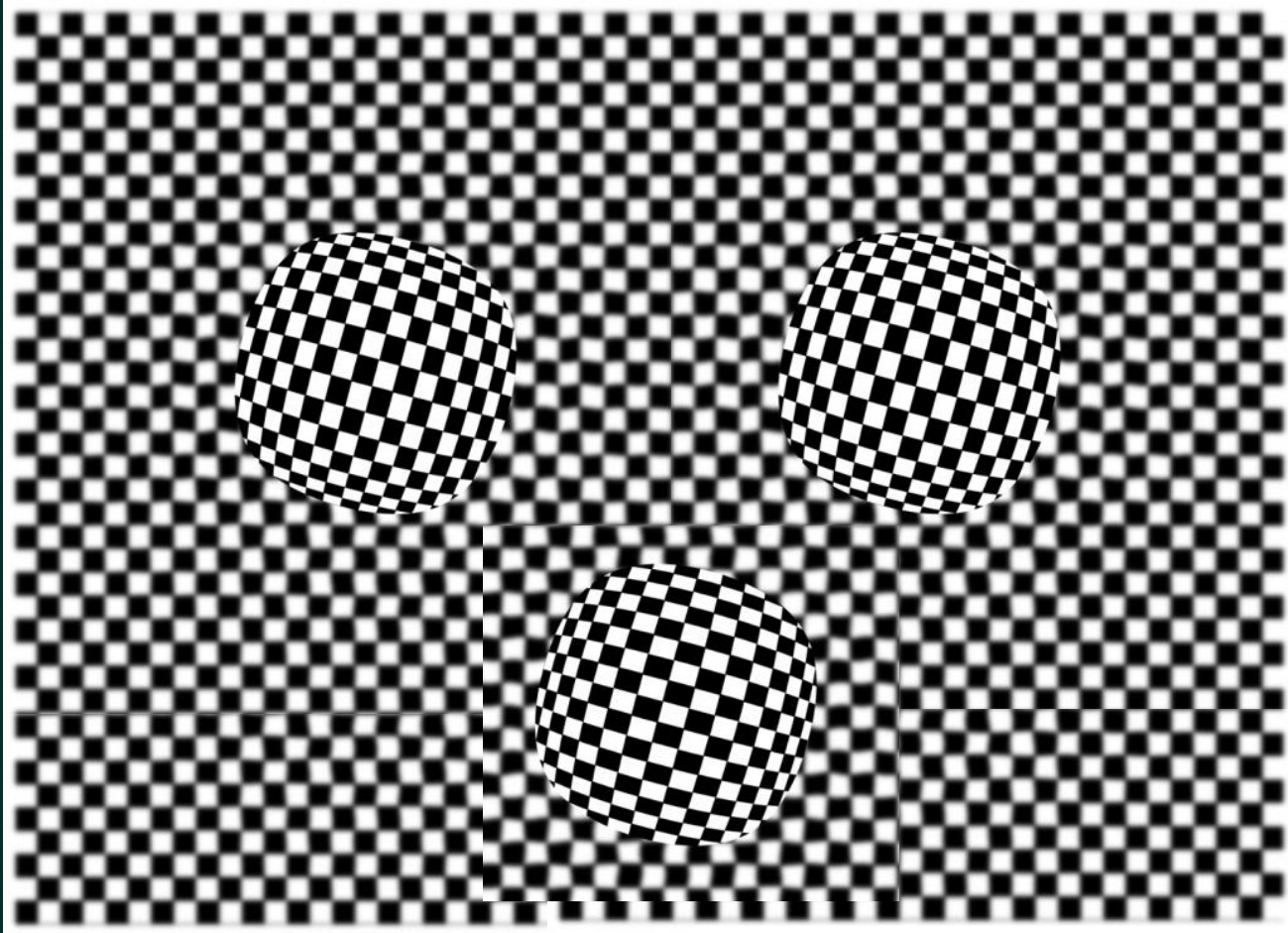


انتزاع عملکرد و ساختار سلسله مراتبی

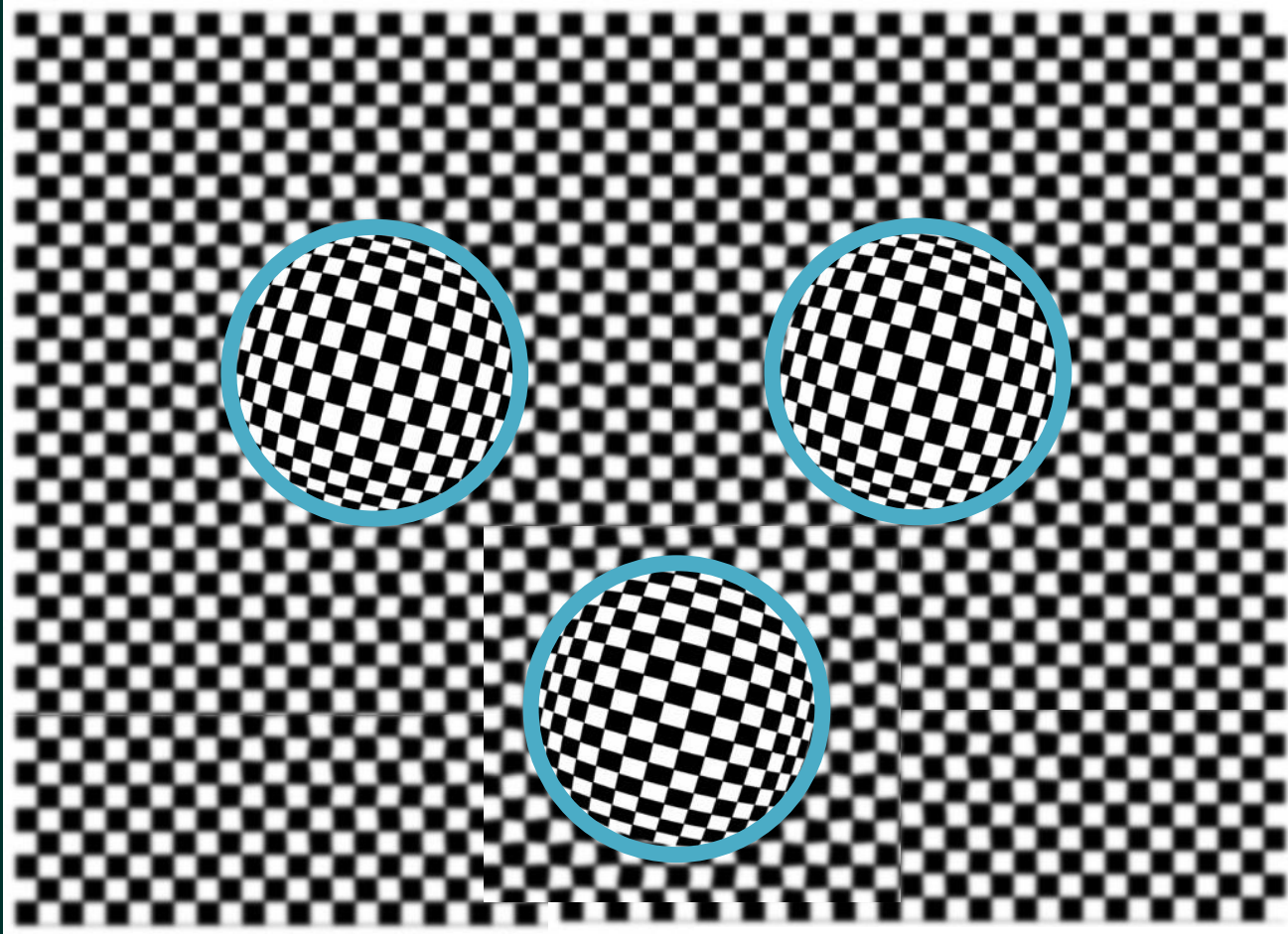
فرض کنید می‌خواهیم با استفاده از مربع‌های زیر نقاشی بکشیم ...



انتزاع عملکرد و ساختار سلسله مراتبی

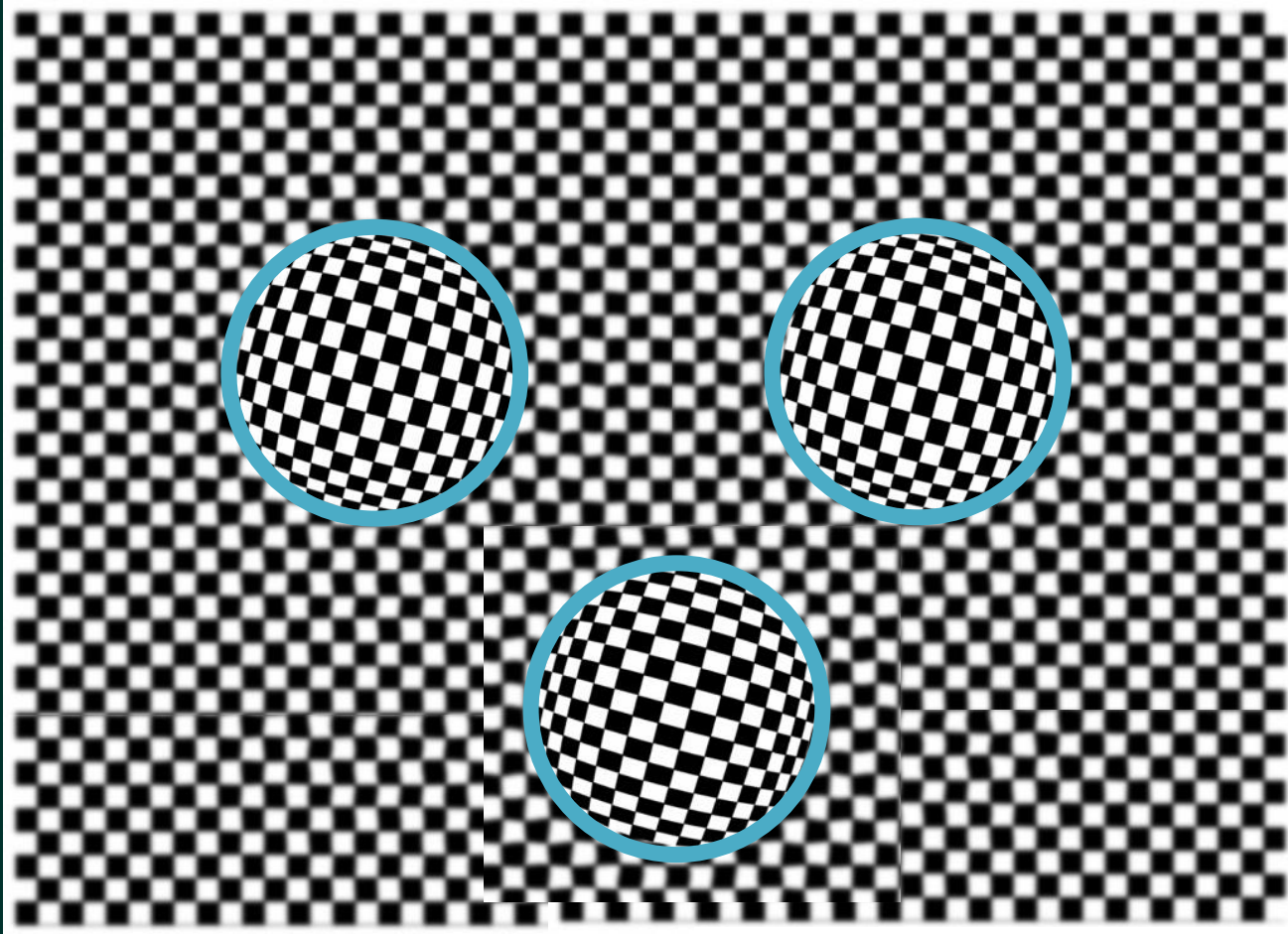


انتزاع عملکرد و ساختار سلسله مراتبی



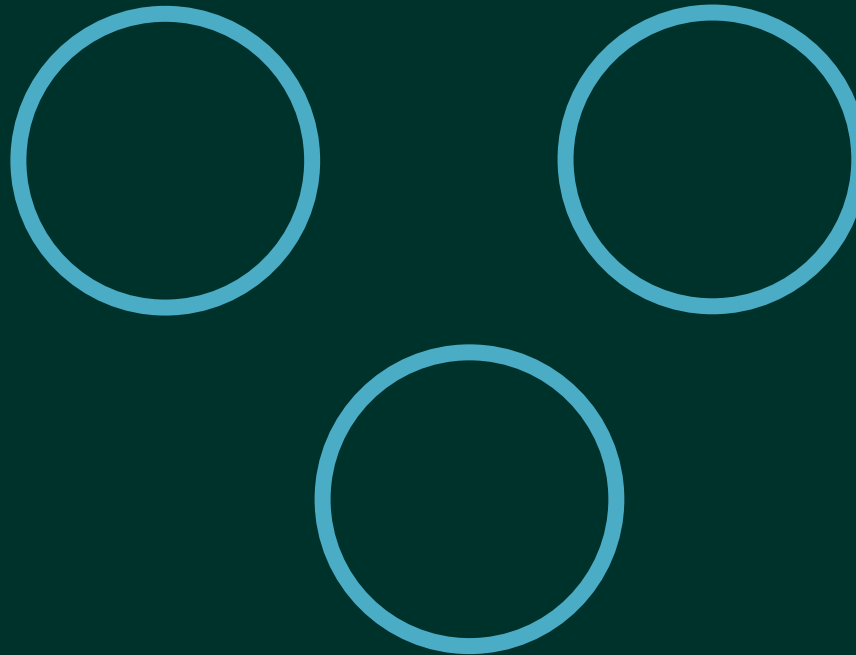
انتزاع عملکرد و ساختار سلسله مراتبی

جزئیات را فراموش کن



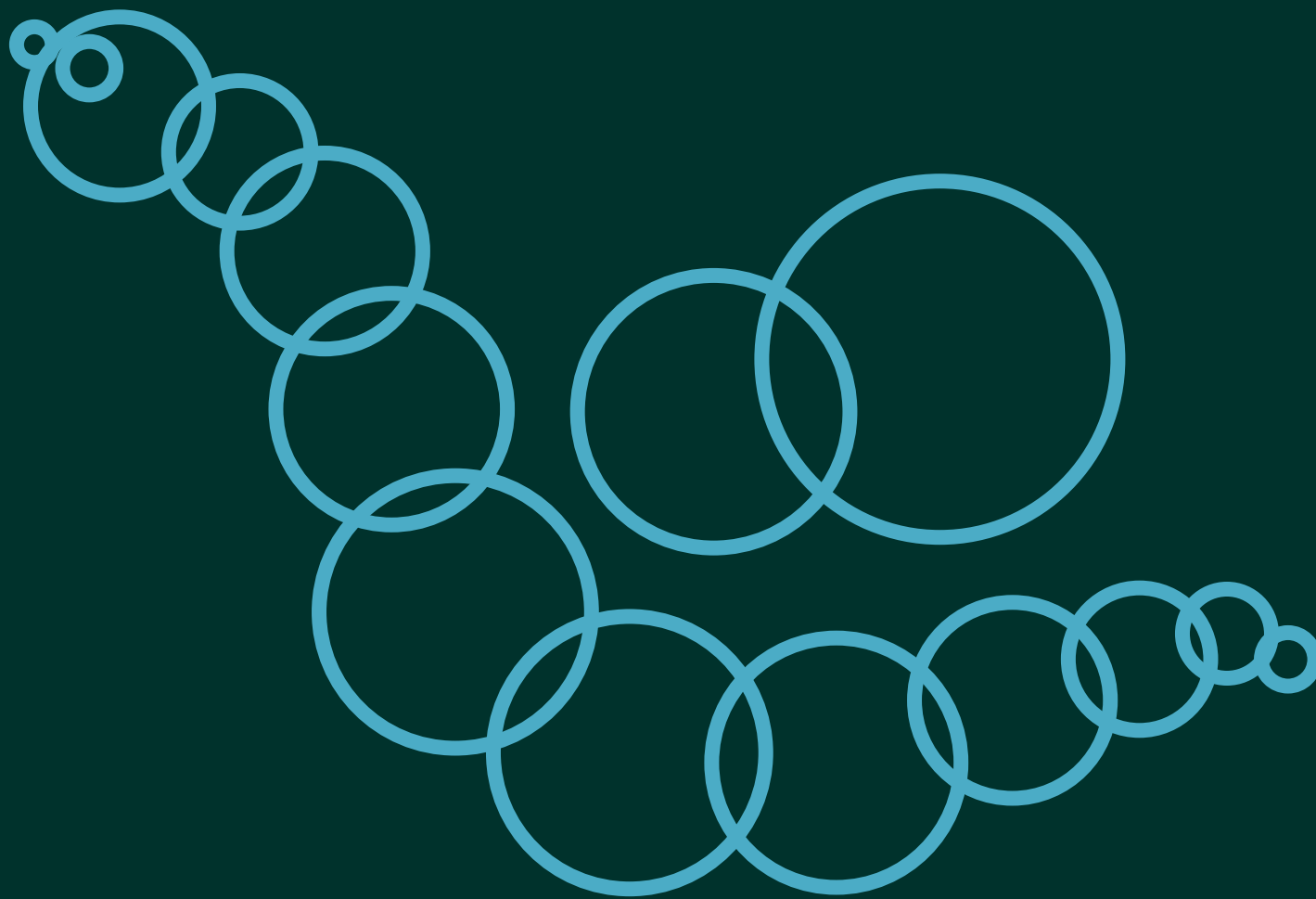
انتزاع عملکرد و ساختار سلسله مراتبی

جزئیات را
فراموش کن
تا نقاشی به
موجودیت‌های
جدید فروکاسته
شود



انتزاع عملکرد و ساختار سلسله مراتبی

جزئیات را
فراموش کن
تا نقاشی به
موجودیت‌های
جدید فروکاسته
شود و بتوانیم
نقاشی‌های
جدید بکشیم



00101010101010101101110110100111101101001010010111010110010010101010101010010
1011101010000001001101101100101000101001011101010010010010101000101000010101000
1001000101011001001010100100111001010101010101011011101101001111011010010100101
1101011001001010101010101010010101110101000000100110110110010100010100101110101
0010010010101000101000010101000100100010101100100101010010011100101010101010101
1011101101001111011010010100101110101100100101010101010101001010111010100000010
0110110110010100010100101110101001001001010100010100001010100010010001010110010
0101010010011100101010101010101101110110100111101101001010010111010110010010101
0101010101001010111010100000010011011011001010001010010111010100100100101010010
0010010001010110010010101001001110010101010101010110111011010011110110100101001
0111010110010010101010101010100101110101000000100110110110010100010100101110101
0010010010101000101000010101000100100010101100100101010010011100101010101010101
1011101101001111011010010100101110101100100101010101010101001010111010100000010
0110110110010100010100101110101001001001010100010100001010100010010001010110010
0101010010011100101010101010101101110110100111101101001010010111010110010010101
0101010101001010111010100000010011011011001010001010010111010100100100101010001
0100001010100010010001010110010010101001001110010101010101010110111011010011110
1101001010010111010110010010101010101010100101011101010000001001101101100101000
1010010111010100100100101010001010000101010001001000101011001001010100100111010

0010101010101010110111011010011110110100101001011101011001001010101010101010010
1011101010000001001101101100101000101001011101010010010010101000101000010101000
1001000101011001001010100100111001010101010101011011101101001111011010010100101
1101011001001010101010101010010101110101000000100110110110010100010100101110101
0010010010101000101000010101000100100010101100100101010010011100101010101010101
101110110100111101101001010010111010110010010101010101010101001010111010100000010
011011011001010001010010111010100100100101010001010100010010001010110010
010101001001110010101010101010101101101010011101101001010010111010110010010101
01010101010010101110101000001001011011001010001010010111010101001001001010101
00100100010101100100101010010011100101010101010101011011011010011110110100101001
011101011001001010101010101010100101110101000000100110110110010100010100101110101
00100100101010001010000010101000100100010101100101010100100111001010101010101
10111011010011110110100101001011101011001010101010101010101010100000010
011011011001010001010010111010100100100101010001010100010010001001001001010110010
010101001001110010101010101010101101110110100111101101001010010111010110010010101
0101010101001010111010100000010011011011001010001010010111010100100100101010001
0100001010100010010001010110010010101001001110010101010101010110111011010011110
110100101001011101011001001010101010101010100101011101010000001001101101100101000
1010010111010100100100101010001010000101010001001000101011001001010100100111010

انتزاع متن از رشته‌های صفر و یک

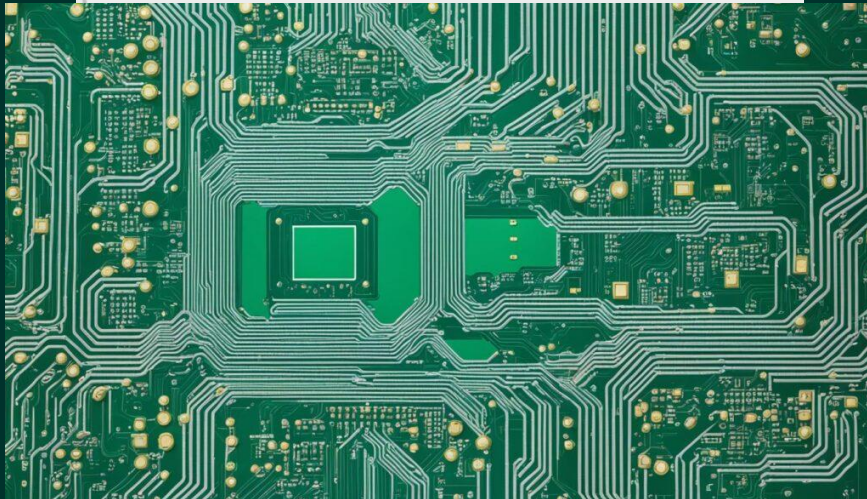
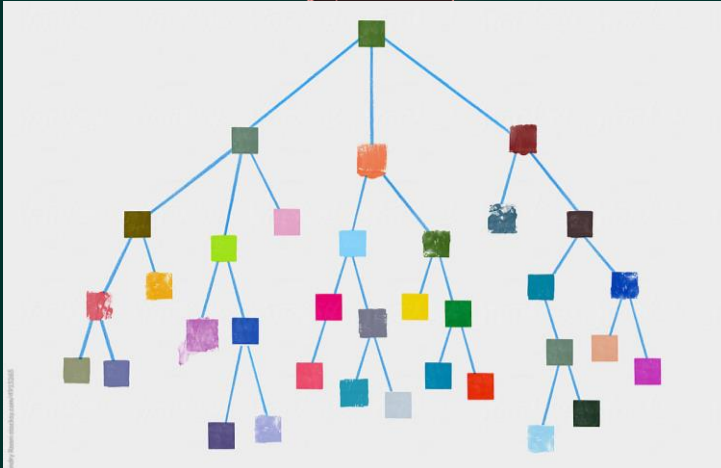
انتزاع متن از رشته های صفر و یک

TEXT

انتزاع زبان از متن

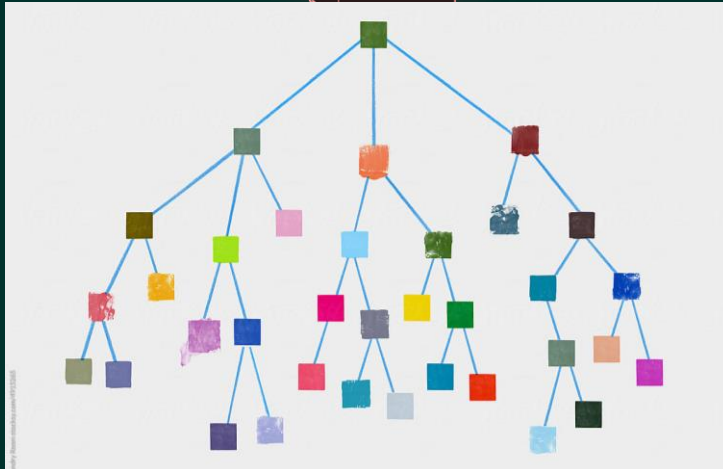
A computer
programming
language is an
abstraction of
texts .

ما چگونه با این
ساختار سلسله‌مراتبی
که در نهایت چیزی
جز محاسبات دودویی
نمی‌فهمد، ارتباط
برقرار می‌کنیم؟





ما چگونه با این
ساختار سلسله‌مراتبی
که در نهایت چیزی
جز محاسبات دودویی
نمی‌فهمد، ارتباط
برقرار می‌کنیم؟



برای پاسخ به این سوال، فعلاً فرض کنید در جهان پیش از
مدل‌های زبانی بزرگ و هوش مصنوعی مولد به سر می‌بریم.

زبان برنامه‌نویسی

- هر برنامه یا هر زبان برنامه نویسی از مجموعه‌ای عملیات ابتدایی (Primitive Operations) تشکیل شده است که به آن‌ها ترکیب یا عبارت (Expression) می‌گویند.
- هر عبارت در زبان برنامه‌نویسی یک ما به‌ازای مقداری دارد (Value) که می‌توان آن را با اغماض مشابه معنی عبارت در زبان انسان در نظر گرفت.

زبان آدم و زبان ماشین

- مولفه‌های تشکیل‌دهنده یک زبان چیست؟

مواد اولیه / خام (set of primitives)

قواعد ترکیب / نحو (means of combination)

ملاحظات معنایی (means of abstraction / static semantics)

زبان برنامه‌نویسی	زبان آدم	
اعداد، رشته‌ها (characters)، عملگرها	کلمات	مواد اولیه
$5 = \text{"salam"} \times$ $x5 = \text{"salam"} \checkmark$	$\times$ من مدرسه شدم دانشگاه وارد از. $\checkmark$ من از مدرسه وارد دانشگاه شدم.	قواعد ترکیب
$5 + \text{"salam"} \times$	$\times$ من از مدرسه وارد دانشگاه شدیم.	ملاحظات معنایی

وقتی قواعد ترکیب (Syntaxes) یا ملاحظات معنایی (Static Semantics) رعایت نشوند اجرای برنامه متوقف می‌شود و ما با **خطا (Error)** مواجه می‌شویم. برای برطرف کردن خطا لازم است پیام خطا را با دقت بخوانیم و برنامه را مطابق فهم کامپیوتر بازتنظیم کنیم. کیفیت پیام‌های خطا در زبان‌های مختلف متفاوت است.

چندمعنایی؛ غیرمجاز برای ماشین

• شاه‌ام شکست ← چندمعنایی در زبان آدم اتفاق می‌افتد

• $5 + 4 * 3$ ← ؟

چندمعنایی؛ غیرمجاز برای ماشین

• شانه‌ام شکست ← چندمعنایی در زبان آدم اتفاق می‌افتد

• $5 + 4 * 3$ ← در زبان برنامه‌نویسی هر عبارت یک معنی (مقدار)

بیشتر ندارد. اما اگر این معنا، همان که منظور ماست
نباشد اتفاقات غیرقابل انتظار می‌افتد

زبان سطح بالا و زبان سطح پایین

یک برنامه کامپیوتری پس از آنکه توسط برنامه‌نویس تکمیل و به کامپیوتر تحویل داده می‌شود با چه سرنوشتی مواجه می‌شود؟

یک برنامه چه مراحل را طی می‌کند تا به یک زبان سطح پایین low-level programming language و در نهایت یک دستورالعمل قابل اجرا برای ماشین تبدیل شود؟

○ ترجمه compile

○ تفسیر interpretation

○ ترکیبی از هر دو در قالب virtual machine



برنامه ورودی input program

پیش‌پردازش preprocess

مترجم Compiler، مفسر Interpreter

زبان اسمبلی Assembly Language

زبان ماشین Machine code

با این مقدمه می‌توانیم با خیال
راحت‌تری سراغ یادگیری زبان‌های
برنامه‌نویسی برویم!