程序设计(Python)

Python基础语法：从思维到实践

主讲：数据与目标工程学院 胡瑞娟 副教授

程序的基本编写方法——IPO模式

I：Input

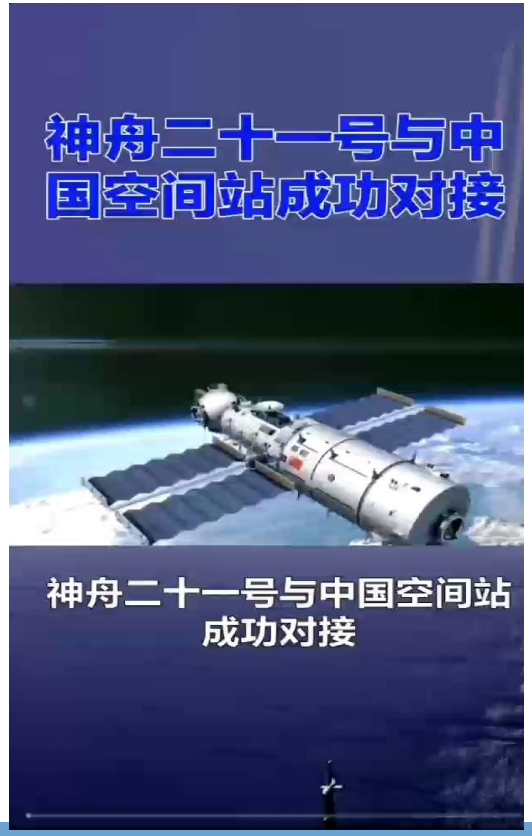
O：Output

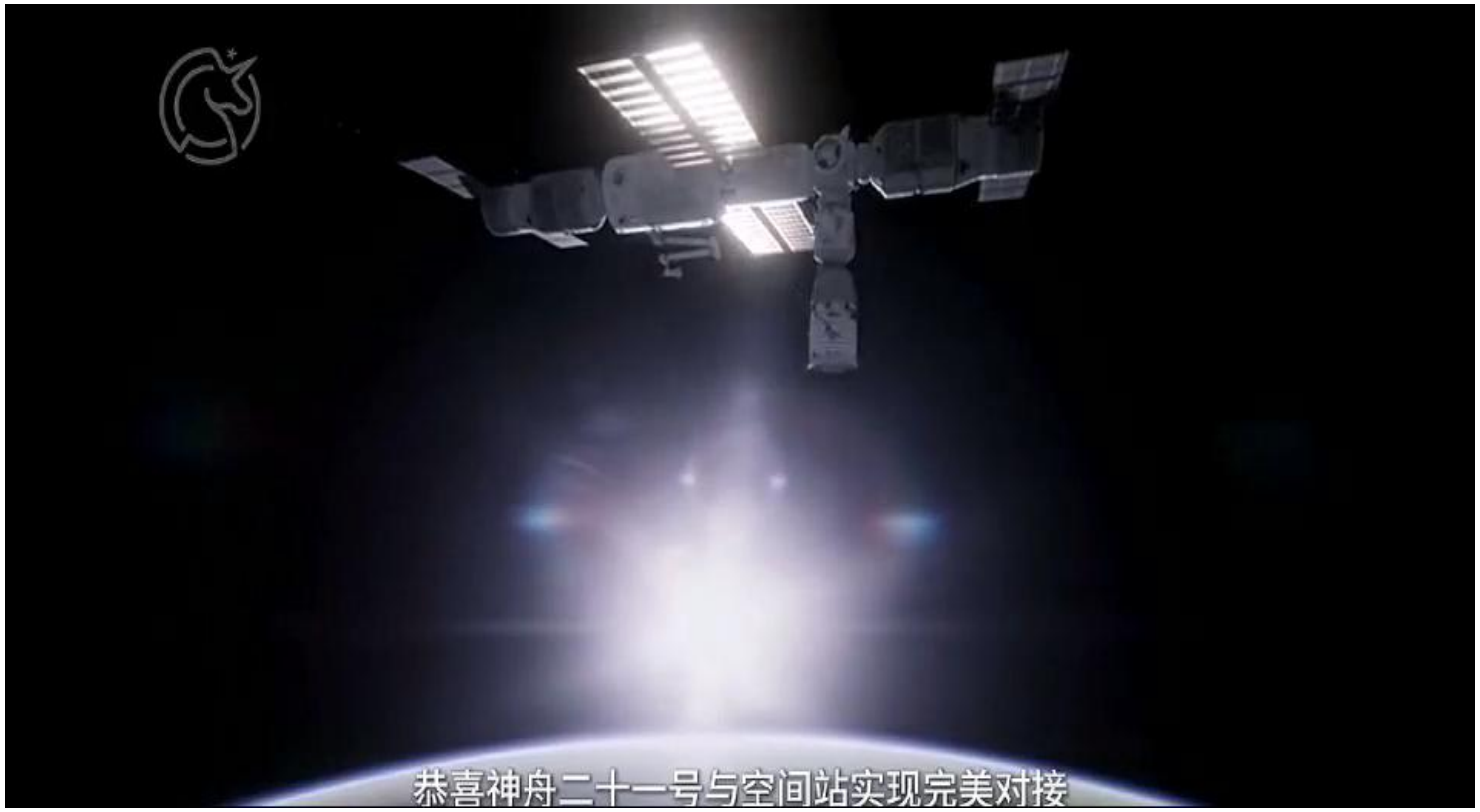
P：Process

计算的过程(算法)

案例引入

【案例】神舟二十一号载人飞船入轨后，于北京时间2025年11月1日3时22分，成功对接于空间站天和核心舱前向端口，整个对接过程历时约3.5小时，创造了神舟飞船与空间站交会对接的最快纪录。11月1日4时58分，在轨执行任务的神舟二十号航天员乘组顺利打开“家门”，欢迎远道而来的神舟二十一号航天员乘组入驻中国空间站，这是中国航天史上第7次“太空会师”。





案例引入

【空间站交会对接模拟】**简化**空间站交会对接过程，用程序模拟。注意：

1.航天员3名，交会**对接耗时**约3.5小时，假设初始距离380公里，计算**平均速度**。

2.燃料消耗估算：即**燃料消耗=基础系数*飞船质量*对接时间**。

3.输出结果如下图。

```
=== 神舟二十一号交会对接模拟 ===
北京时间2025年11月1日3时22分，神舟二十一号与天宫空间站成功对接。
【任务基础数据】
飞船名称： 神舟二十一号
飞船质量： 8.0 吨
航天员人数： 3 人
距离：380.0公里
官方报道对接耗时： 3.5 小时

【对接过程关键计算】
1. 平均接近速度： 108.57 公里/小时，
2. 估算对接阶段燃料消耗： 0.42 吨
```

```
=== 神舟二十一号交会对接模拟 ===
北京时间2025年11月1日3时22分，神舟二十一号与天宫空间站成功对接。
输入飞船名称： 神州21号
输入飞船质量： 8
输入对接时间： 3.5
距离： 380
航天员人数： 3
【任务基础数据】
飞船名称： 神州21号
飞船质量： 8.0 吨
航天员人数： 3 人
距离：380公里
官方报道对接耗时： 3.5 小时

【对接过程关键计算】
1. 平均接近速度： 108.57 公里/小时，
2. 估算对接阶段燃料消耗： 0.42 吨
```

案例引入

【空间站交会对接模拟】**简化**空间站交会对接过程，用程序模拟。注意：

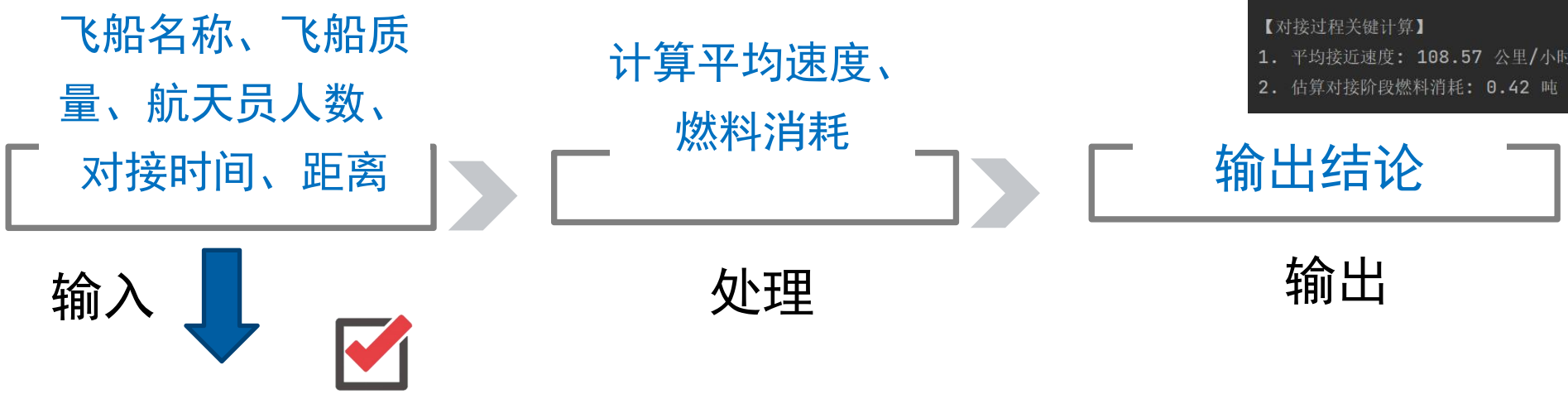
1.航天员3名，交会**对接耗时**约3.5小时，假设初始距离380公里，计算**平均速度**。

2.燃料消耗估算：即**燃料消耗=基础系数*飞船质量*对接时间**。

```
=== 神舟二十一号交会对接模拟 ===
北京时间2025年11月1日3时22分，神舟二十一号与天宫空间站成功对接。
【任务基础数据】
飞船名称：神舟二十一号
飞船质量：8.0 吨
航天员人数：3 人
官方报道对接耗时：3.5 小时

【对接过程关键计算】
1. 平均接近速度：108.57 公里/小时
2. 估算对接阶段燃料消耗：0.42 吨
```

【案例分析】IPO方法



思考重点：1. 需要几个数据？2. 用**什么**来表示数据？3. **数据的类型**是？

4. 数据的来源是（固定值、**从键盘输入**……. .）？

主要内容

01 变量：数据的“容器”

02 数据类型

03 运算符与表达式

04 输出函数

01 变量：数据的“容器”

2.变量：数据的“容器”

【问题2】用什么表示数据?

• 飞船名称	• name	• spacecraft_name
• 飞船质量	• mass	• ship_mass_ton
• 航天员人数	• number	• crew_members
• 对接时间	• time	• dock_time
• 距离	• distance	• distance_km

变量

2.变量：数据的“容器”

➢ 变量

- **变量**：代表存储在计算机存储器中的某个值的**名字**，程序执行过程中，值发生改变或需要改变的元素。
- **命名规则**: 大小写字母、数字、下划线和中文等字符及组合

  如: TempStr, Python_Great, 这是门Python好课
- 注意: **大小写敏感、首字符不能是数字、不与关键字相同**，Python和python是不同变量，123Python是不合法的

2.变量：数据的“容器”

Python 关键字（Python语言的关键组成部分）

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

```
>>> help('keywords')

Here is a list of the Python keywords.  Enter any keyword to get more help.

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

>>> _
```

- 注意：关键字几乎全部小写，且**不能用作标识符（变量名、函数名、类名）**

【习题】下列变量名中不合法的是：（）

A. a　　B. _a　　C. _a2　　D. 2a

【习题】下列变量名中不合法的是：（）

A. name

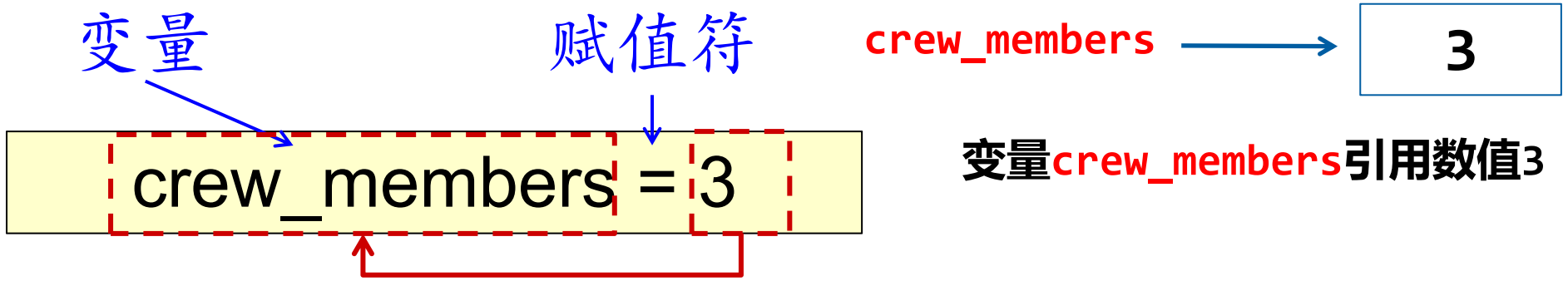
B. apple_price

C. num

D. *age

2.变量：数据的“容器”

- 赋值



变量crew_members引用数值3

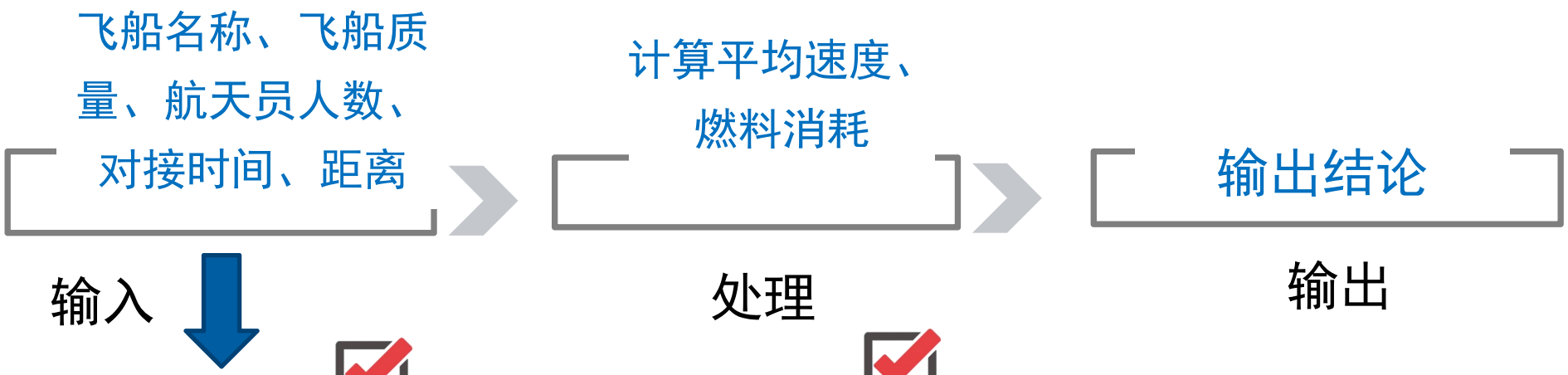
先计算右边的值，再赋给左边的变量

- 例：

```
ship_mass_ton=8.0
ship_mass_ton = ship_mass_ton - 0.5
```

先计算ship_mass_ton-0.5的值(7.5)，再将7.5赋给左边的变量a

2.变量：数据的“容器”



思考重点：1. 需要几个数据？2. 用**什么**来表示数据？3. **数据的类型**是？

4. 数据的来源是（固定值、**从键盘输入**……..）？

思考：不同数据类型的表示方法？

变量

- spacecraft_name = "神舟二十一号"
- ship_mass_ton = 8.0
- crew_members = 3.5
- dock_time = 380.0
- distance_km = 3

02

数据类型

3. 数据类型

- Python数据类型分为：*基本数据类型*和组合数据类型
  - 基本数据类型：数字类型（整数（int）、浮点数（float））
    布尔类型（bool）
    字符串类型(str)
  - 组合数据类型：集合、元组、列表、字典

不同数据类型的表示方法

- a=10 #整数
- b= “10” #字符串
- c= ‘c’ #字符串
- d=True #布尔型
- add=3+4
- add=a+b

表达式

查看变量的数据类型的函数

type(变量)

```
>>> a=10
>>> print(a)
10
>>> print(type(a))
<class 'int'>
```

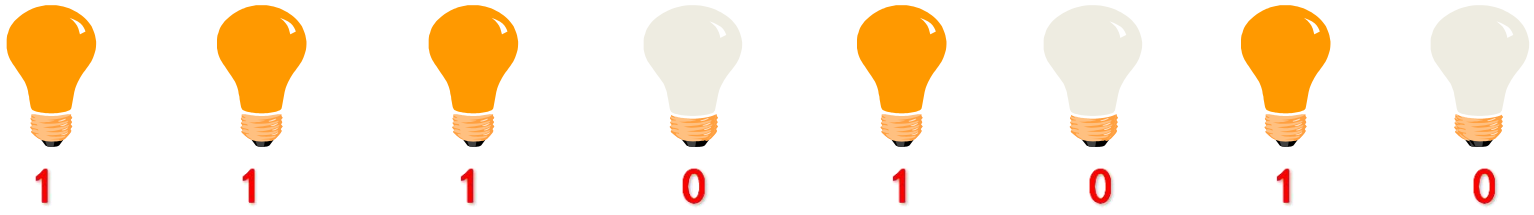
```
>>> b="10"
>>> print(b)
10
>>> print(type(b))
<class 'str'>
```

对象类型	示例
数字	1234, 3.14, 3+4j
字符串	'swfu', "I'm student", '"Python "'
列表	[1, 2, 3, '3']
字典	{1:'food' ,2:'taste', '3':'import'}
元组	(2, -5, 6, '3')
文件	f=open('data.dat', 'r')
集合	set('abc'), {'a', 'b', 'c'}
布尔型	True, False
空类型	None
编程单元类型	函数、模块、类

【思考】为什么需要区分这么多类型?

■因为在计算机的底层，所有数据——无论是数字、文字还是图像——最终都会被转换成由 0 和 1 组成的二进制码来存储和处理。

■可以通过不同的编码方式来表示不同类型的数据(字符：ASCII，浮点数：IEEE标准，图像：位图等）。



✓ 只有“0”和“1”两个数字

- 121是一个二进制数吗?

✓ 按“逢二进一”的规则计数：1+1=10

我认识它已有10110年了。

数据的存储单位

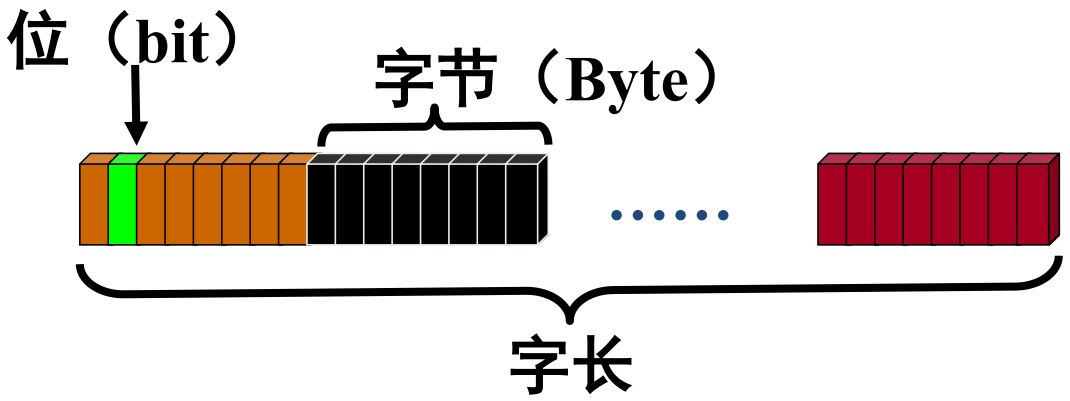
位(bit)：计算机存储数据的最小单元(0、1)

字节(Byte)：处理数据的基本单位(8bit/Byte)

常用的字节计数单位：

1KB＝1024 Byte $(2^{10}B)$　　1MB＝1024 KB $(2^{20}B)$

1GB＝1024 MB $(2^{30}B)$　　1TB＝1024 GB $(2^{40}B)$



字长：CPU一次处理数据的二进制位数。

数据类型——整数

- 数制的概念

常用数制	十进制	二进制	八进制	十六进制
数字符号	0～9	0，1	0～7	0～9, A, B, C, D, E, F
基 数	10	2	8	16

基数：R进制的基数=R

位权：是一个与数字位置有关的常数，位权=R^n

其中n取值：以小数点为界，向左 0，1，2，3……，
向右-1，-2，-3……

常用数制的对应关系

十进制	二进制	八进制	十六进制
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10

- 二进制的运算

二进制的算术运算

加：	减：	乘：	除：
0+0=0	0-0=0	0×0=0	0÷0=0
0+1=1	0-1=1	0×1=0	0÷1=0
1+0=1	1-0=1	1×0=0	1÷0(无意义)
1+1=10	1-1=0	1×1=1	1÷1=1

本位为0，
向高位进位1

本位为1，
向高位借1当2

- 二进制的运算

二进制的逻辑运算

0表示“假、否”，1表示“真、是”

与AND：

$0 \wedge 0 = 0$

$0 \wedge 1 = 0$

$1 \wedge 0 = 0$

$1 \wedge 1 = 1$

或OR：

$0 \vee 0 = 0$

$0 \vee 1 = 1$

$1 \vee 0 = 1$

$1 \vee 1 = 1$

非(取反)：

$\overline{0} = 1$

$\overline{1} = 0$

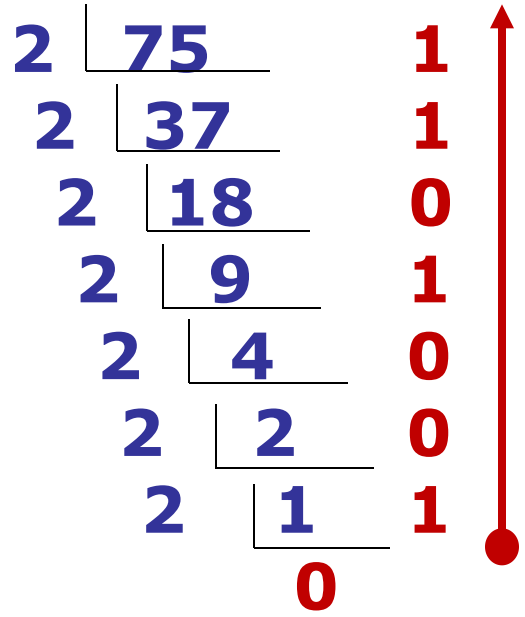
$\wedge$表示串联，$\vee$表示并联

- 进制的转换

十进制 ➡ 二进制、八进制、十六进制

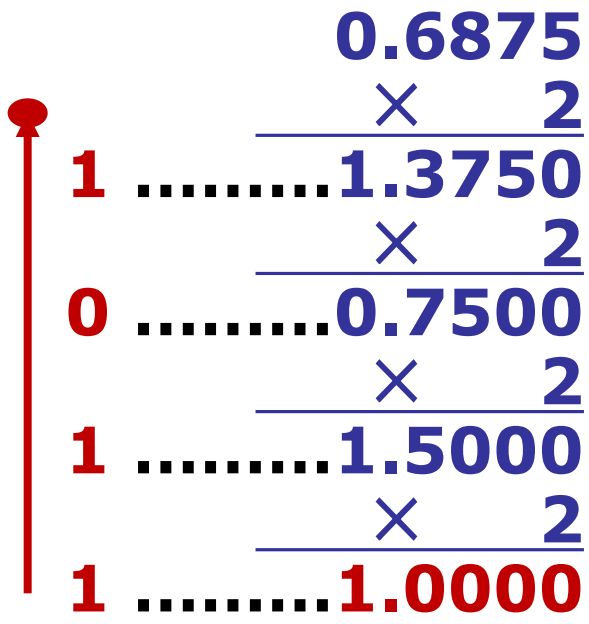
- 十进制转二进制：整数部分除以2取余，直至商为0；小数部分乘以2取整，直至小数部分为0或达到所需精度为止。
- 十进制转八进制：方法同上。整数部分除以8，小数部分乘以8。
- 十进制转十六进制：方法同上。整数部分除以16，小数部分乘以16。

十进制整数➡二进制整数



结果为：1001011

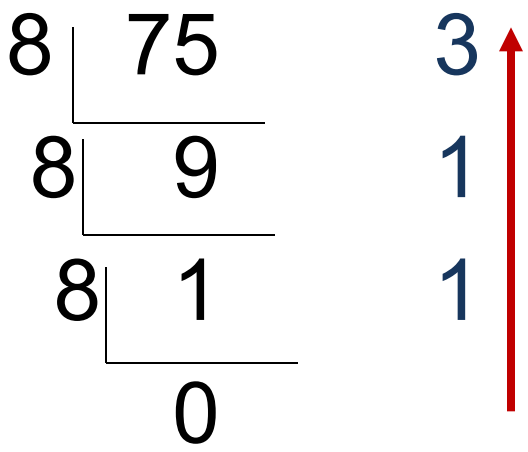
十进制小数➡二进制小数



结果为：0.1011

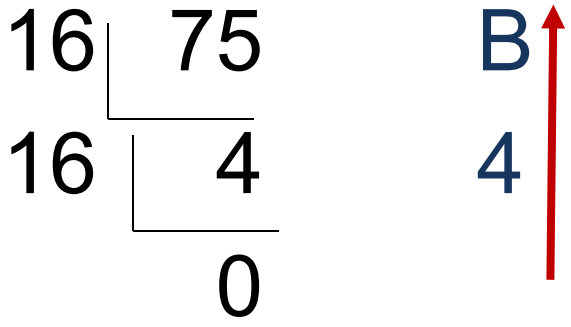
$(75.6875)_{10}=(\qquad)_2$

十进制整数➡八进制整数



结果为：113

十进制整数➡十六进制整数



结果为：4B

二进制、八进制、十六进制 ⟶ 十进制

***位权相加法*：各位数码乘位权，再相加。**

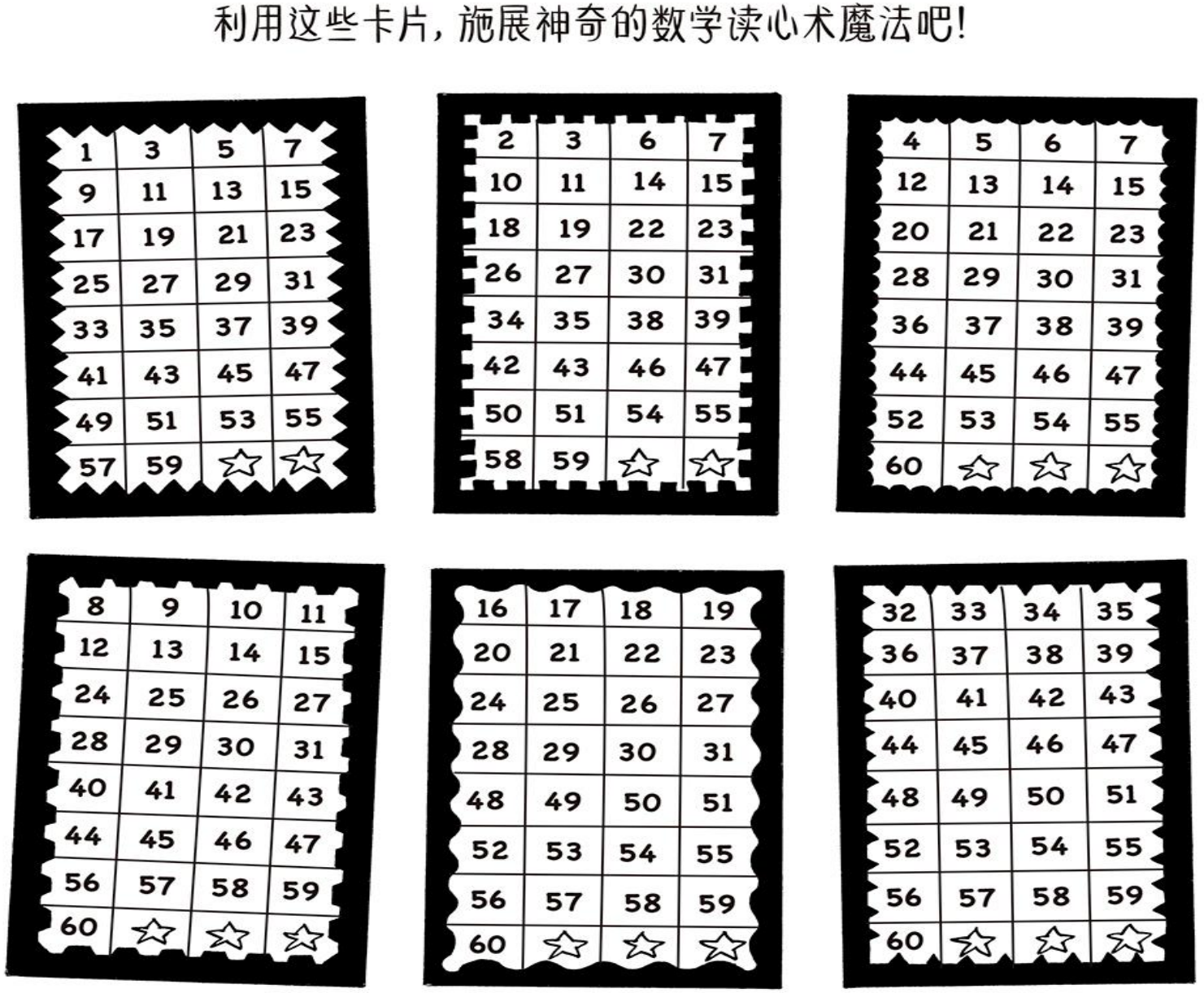
例：

$$(1011.1)_2 = 1\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0 + 1\times2^{-1}$$

$$= 8 + 0 + 2 + 1 + 0.5$$

$$= (11.5)_{10}$$

- 游戏：数字读心术
- 你说，我来猜



数据类型——浮点数

- 浮点数又称小数

15.0、0.37、-11.2、1.2e2、314.15e-2

```
>>> 0.2+0.1
0.30000000000000004
>>> 2.1-2.0
0.10000000000000009
```

print(0.1 + 0.2)　# 不确定尾数问题

round(x , b)：对x四舍五入，b是保留小数的位数

print(round(0.1 + 0.2,1)　# 消除不确定尾数

（1）数字类型——定义与赋值

- 整数 (int)
  - a=123
  - b=0b101　　# 二进制
  - c=0o017　　#八进制
  - d=0x2F　　#十六进制
- 浮点数 (float)
  - f=1.23
  - d=1.7e-5　　(等价于$1.7*10^{-5}$,即0.000017)

（2）输入函数

思考：除了直接给变量赋值，如何从键盘输入值呢？

• spacecraft_name	= "神舟二十一号"
• ship_mass_ton	= 8.0
• crew_members	= 3.5
• dock_time	= 380.0
• distance_km	= 3

```
=== 神舟二十一号交会对接模拟 ===
北京时间2025年11月1日3时22分，神舟二十一号与天宫空间站成功对接。
【任务基础数据】
飞船名称：神舟二十一号
飞船质量：8.0 吨
航天员人数：3 人
距离：380.0公里
官方报道对接耗时：3.5 小时
```

```
=== 神舟二十一号交会对接模拟 ===
北京时间2025年11月1日3时22分，神舟二十一号与天宫空间站成功对接。
输入飞船名称：神州21号
输入飞船质量：8
输入对接时间：3.5
距离：380
航天员人数：3
【任务基础数据】
飞船名称：神州21号
飞船质量：8.0 吨
航天员人数：3 人
距离：380公里
官方报道对接耗时：3.5 小时
```

（2）输入函数

从控制台获得用户输入的函数

- **input()函数的使用格式：**

```
<变量> = input([提示信息字符串])
```

- **用户输入的信息以字符串类型保存在<变量>中**

```
number1=input() # number保存用户输入的信息，无提示信息 TempStr =
input("请输入") # TempStr保存
```

字符串类型

如何验证变量的数据类型？

type(变量)

问题：如果用户需要的不是字符串类型怎么办？

（3）类型转换——数值运算内置函数

数据类型转换

函数及使用	描述
int(x)	将x变成整数，舍弃小数部分 int(123.65) 结果为123； int("123") 结果为123 int(“123.45”)程序报错
float(x)	将x变成浮点数，增加小数部分 float(12) 结果为12.0； float("1.23") 结果为1.23

```
a=int(input())
```

整型

```
b=float(input())
```

浮点型

单选题 1分

设置

number1=input()，若从键盘输入2，则number1的值为（ ），数据类型为（）。

A 2，数字类型

B “2”，字符串

C 2，字符串

D “2”，数字类型

提交

单选题 1分

设置

```
a=input()
b=float(input())
```

从键盘输入14和15后，a的值为（　），b的值为（　）

A　“14”　15.0

B　14　15

C　14.0　15.0

D　14.0　15.0

提交

（4）评估函数eval()

去掉参数最外侧引号并执行余下语句的函数

- **eval()函数的基本使用格式：**

```
<变量>= eval(<字符串或字符串变量>)
```

```
>>> eval("1")
1
>>> eval("1+2")
3
```

```
a=eval( input() )
```

是什么数据类型呢？

单选题 1分

设置

```
a=eval(input())
b=eval(input())
```

从键盘输入14和15.4后，a的类型为（），b的类型为（）

A 字符串 字符串

B 整型 浮点型

C 整型 整型

D 浮点型 整型

提交

3. 数据类型

思考重点：1. 需要几个数据？2. 用**什么变量名**来表示数据？3. **数据的类型**是？

4. 数据的来源是（固定值、**从键盘输入**……..）？

```
spacecraft_name = input("输入飞船名称：")
ship_mass_ton = float(input("输入飞船质量："))
dock_time = eval(input("输入对接时间："))    # 浮点数：对接耗时3.5小时
distance_km = eval(input("初始距离："))  # 浮点数：初始距离，假设值，单位公里
crew_members = eval(input("航天员人数："))   # 整数：航天员人数
```

平均速度=距离/对接时间

```
average_speed_kmh = distance_km / dock_time
```

燃料消耗=基础系数*飞船质量*对接时间

```
fuel_consumption_factor = 0.015
fuel_used = fuel_consumption_factor * ship_mass_ton *
dock_time
```

03 运算符与表达式

数字类型的操作——算术运算符

运算符	描述	实例（a=10,b=20）
+	加	a + b 结果为 30
-	减	a - b 结果为 -10
*	乘	a * b 结果为 200
/	除	b / a 结果为 2
%	取模（求余）	b % a 结果为 0
**	幂	a**b 结果为10^{20}
//	整除	9//2 结果为 4

单选题 1分

设置

n1=9//2
n2=9%2
则n1、n2的值分别为（　）

A　4.5　4

B　4　1

C　1　4

提交

4. 运算符与表达式

运算符种类分类

Python运算符可分为七大类，覆盖数学计算、逻辑判断、对象操作等场景：

1. **算术运算符**：执行数学运算（+、−、*、/、//、%、**）
2. **赋值运算符**：变量赋值与运算结合（=、+=、−=等）
3. **比较运算符**：比较值大小或内容（==、!=、>、<等）
4. **逻辑运算符**：组合布尔值（and、or、not）
5. **位运算符**：二进制位操作（&、|、^、<<、>>）
6. **成员运算符**：判断元素是否在序列中（in、not in）
7. **身份运算符**：判断对象内存地址是否相同（is、is not）

判断元素是否在序列中：`"a" in ["a","b"] → True`

否定形式：`"c" not in ("a","b") → True`

判断内存地址是否相同：`a = [1]; b = a; a is b → True`

否定形式：`a = [1]; b = [1]; a is not b → True`

优先级规则解析

运算符优先级从高到低分为以下层级（同级按结合性处理）：

1. **括号()**：强制改变运算顺序
2. **幂运算(**)**：右结合，2**3**2 → 2**(3**2)=512
3. **一元运算符（+、−、~）**：−5 + 3 → (−5)+3=−2
4. **乘除/取模/整除（*、/、%、//）**：左结合，8/2/2 → (8/2)/2=2.0
5. **加减（+、−）**：左结合，5−3+2 → (5−3)+2=4
6. **位移（<<、>>）**：左结合，4<<1>>1 → (4<<1)>>1=4
7. **比较运算（==、!=、>、<等）**：链式比较支持，1
8. **逻辑非（not）**：not True → False
9. **逻辑与（and）**：True and False → False
10. **逻辑或（or）**：False or True → True
11. **赋值运算（=、+=等）**：右结合，a=b=5等价于a=(b=5)

注：优先级规则可通过()显式调整，避免歧义。

3. 数据类型

```
average_speed_kmh = distance_km / dock_time
fuel_consumption_factor = 0.015
fuel_used = fuel_consumption_factor * ship_mass_ton * dock_time
```

飞船名称、飞船质量、航天员人数、对接时间、距离

计算平均速度、燃料消耗

输出结论

输入 → 处理 → 输出

思考重点：需要几个数据？用**什么变量名**来表示数据？**数据的类型**是？

数据的来源是（固定值、**从键盘输入**……..）？

```
spacecraft_name = input("输入飞船名称：")
ship_mass_ton = float(input("输入飞船质量："))
dock_time = eval(input("输入对接时间："))    # 浮点数：对接耗时3.5小时
distance_km = eval(input("初始距离："))  # 浮点数：初始距离，假设值，单位公里
crew_members = eval(input("航天员人数："))  # 整数：航天员人数
```

04

输出函数

输出函数示例：

程序

```
data_1=2
data_2="my name is rjj"
data_3=True
data_4=3
c="A"
print(data_1,data_2,data_1+data_4,c,6,"c")
```

用逗号隔开多个变量、常量或表达式

表达式

运行结果

```
2 my name is rjj 5 A 6 c
```

输出结果中用空格隔开多个变量和常量

print(f' ')进行格式化输出

```
name = 'Alex'
job = 'Software Developer'
# 使用print(f'{}')格式化输出
print(f'My name is {name}. I am a {job}.')
```

以字符串的形式输出

使用方法：大括号表示槽，在槽中写入要输出的变量名
输出时将槽中变量对应的值输出到槽所在的位置

输出结果

```
My name is Alex. I am a Software Developer.
```

单选题 1分

设置

e=123
print(e,“e”)
运行结果为（ ）

A 123，e

B “123”，e

C 123 e

D 123 e

提交

完整程序？

```
print("=== 神舟二十一号交会对接模拟 ===")
print("北京时间2025年11月1日3时22分，神舟二十一号与天宫空间站成功对接。")
# 1. 变量与数据类型 - 存储任务基本数据
spacecraft_name = input("输入飞船名称：")
ship_mass_ton = float(input("输入飞船质量："))
dock_time = eval(input("输入对接时间："))   # 浮点数：对接耗时3.5小时
distance_km = eval(input("距离："))  # 浮点数：距离，假设值，单位公里
crew_members = eval(input("航天员人数："))  # 整数：航天员人数

print("【任务基础数据】")
print(f"飞船名称：{spacecraft_name}")
print(f"飞船质量：{ship_mass_ton} 吨")
print(f"航天员人数：{crew_members} 人")
print(f"距离：{distance_km}公里")
print(f"官方报道对接耗时：{dock_time} 小时\n")

# 2. 运算符与表达式 - 计算关键参数
print("【对接过程关键计算】")
average_speed_kmh = distance_km / dock_time
fuel_consumption_factor = 0.015
fuel_used = fuel_consumption_factor * ship_mass_ton * dock_time
print(f"1. 平均接近速度：{average_speed_kmh:.2f} 公里/小时，2. 估算对接阶段燃料消耗：{fuel_used:.2f} 吨")
```

```
=== 神舟二十一号交会对接模拟 ===
北京时间2025年11月1日3时22分，神舟二十一号与天宫空间站成功对接。
输入飞船名称：神州21号
输入飞船质量：8
输入对接时间：3.5
距离：380
航天员人数：3
【任务基础数据】
飞船名称：神州21号
飞船质量：8.0 吨
航天员人数：3 人
距离：380公里
官方报道对接耗时：3.5 小时

【对接过程关键计算】
1. 平均接近速度：108.57 公里/小时，
2. 估算对接阶段燃料消耗：0.42 吨
```

完整程序-输出结果

```
【任务基础数据】
飞船名称： 神州21号
飞船质量： 8.0 吨
航天员人数： 3 人
距离：380公里
官方报道对接耗时： 3.5 小时

【对接过程关键计算】
1. 平均接近速度： 108.57 公里/小时，2. 估算对接阶段燃料消耗： 0.42 吨
```

如何实现字符串的换行？

换行

在Python中，可以通过以下几种方式实现换行输出：

1. 使用 `\n` 换行符

在字符串中插入 `\n` 可以实现换行。

```python
print("第一行\n第二行")
```

输出：

```
第一行
第二行
```

2. 使用 `print()` 函数

`print()` 函数默认会在输出结束时换行。你可以通过多次调用 `print()` 来实现换行。

```python
print("第一行")
print("第二行")
```

输出：

```
第一行
第二行
```

3. 使用多行字符串

使用三引号（`'''` 或 `"""`）可以创建多行字符串。

```python
print('''第一行
第二行''')
```

输出：

```
第一行
第二行
```

思考与讨论

现代空战中，掌握飞机的性能数据很重要，你来扮演航空数据分析师，为空军建立一套飞机性能数据管理系统。我们需要记录各种战斗机的性能参数，并进行相关的计算分析。为运-20运输机创建数据档案如下，并计算 1. 长宽比（机长/翼展），北京到广州的飞行时间（距离约1900公里）。

```
# 型号："运-20"
# 翼展: 50米
# 机长: 47米
# 机高: 15米
# 最大航程: 7800公里
# 最大速度: 920公里/小时
```

总结

- ☑ 变量定义 – 为数据赋予有意义的名字
- ☑ 数据类型 – 区分文本、数字、真假值
- ☑ 运算符 – 进行数学计算和逻辑操作
- ☑ 输入输出 – 与用户交互，展示结果

提醒：每一个数据都关系到国家、军事安全。我们写代码要严谨细致，确保每个变量、每次计算都准确无误！培养严谨的军事数据意识、树立精确计算的科学态度、增强国防科技自豪感！

学会“分析-建模-解决”！

作业：天天向上的力量。

- 1951年，毛泽东主席题词“好好学习，天天向上”，成为激励一代代中国人奋发图强的经典语录，可是在实际操作中难免会让人疑惑，“好好学习”究竟能好到什么程度呢？

作业：天天向上的力量。

初始值为1，一年之后的值？

每天进步	最终值	每天退步	最终值
0. 1%		0. 1%	
0. 5%		0. 5%	
1%		1%	
周内1%，周末1%			

感悟坚持的力量！

初始值为1，一年之后的值？

进步	最终值	退步	最终值
0. 1%	1. 44	0. 1%	0. 69
0. 5%	6. 17	0. 5%	0. 16
1%	37. 78	1%	0. 03
周内1%，周末1%	4. 63		

感悟坚持的力量！

授课结束

下课并不代表思考的终止~

而是对我们下一次交流的期盼~