

分枝， 迭代

（下载幻灯片和 .py 文件 Ā Ē 跟随！）

6.0001 讲座 2

上次

语法和语义

标量对象

简单的操作

表达式、变量和值

今天

字符串对象类型

分支和条件

缩进

迭代和循环

字符串

字母、特殊字符、空格、数字

用引号或单引号括起来

hi = “你好”

连接字符串

名称= “安娜”

问候=嗨+名字

问候=嗨+ + 姓名

对 Python 文档中定义的字符串执行一些操作

傻=嗨+ + 姓名 * 3

输入/输出:打印

用于向控制台输出内容

关键字是打印

输入/输出:输入 (“ ”)

打印引号中的任何内容用户输入内容

并点击回车

将该值绑定到一个变量

```
text = input( 输入任何内容... )
```

打印 (5*文本)

输入给你一个字符串,所以如果使用数字则必须强制转换

```
num = int(input( 输入一个数字... ))
```

打印 (5*个)

int、float、string 上的比较运算符

i 和 j 是变量名

下面的比较评估为布尔值

我 > j

我 >= j

我 < j

我 <= j

i == j 等式测试,如果 i 与 j 相同则为 True i != j 不等式测试,如果 i 与 j 不同则为 True

布尔逻辑运算符

a 和 b 是变量名（带有布尔值）

not a True 如果 a 为 False
 如果 a 为真,则为假

a 和 b 如果两者都为真,则为真

a 或 b 如果其中一个或两个都为真,则为真

一个	乙	甲和乙	甲或乙
真的	真的	真的	真的
真的	错误的	错误的	真的
错误的	真的	错误的	真的
错误的	错误的	错误的	错误的

比较示例

pset_time = 15

睡眠时间 = 8

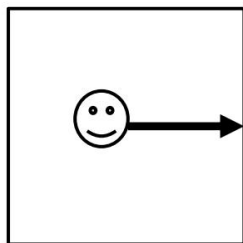
打印 (sleep_time > pset_time)

派生=真

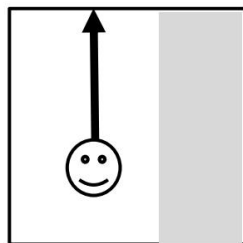
喝=假

两者 = 喝和衍生

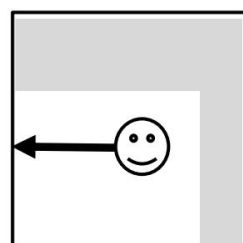
打印 (两者)



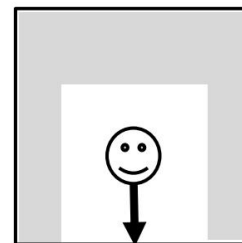
如果清楚的话，
向右走



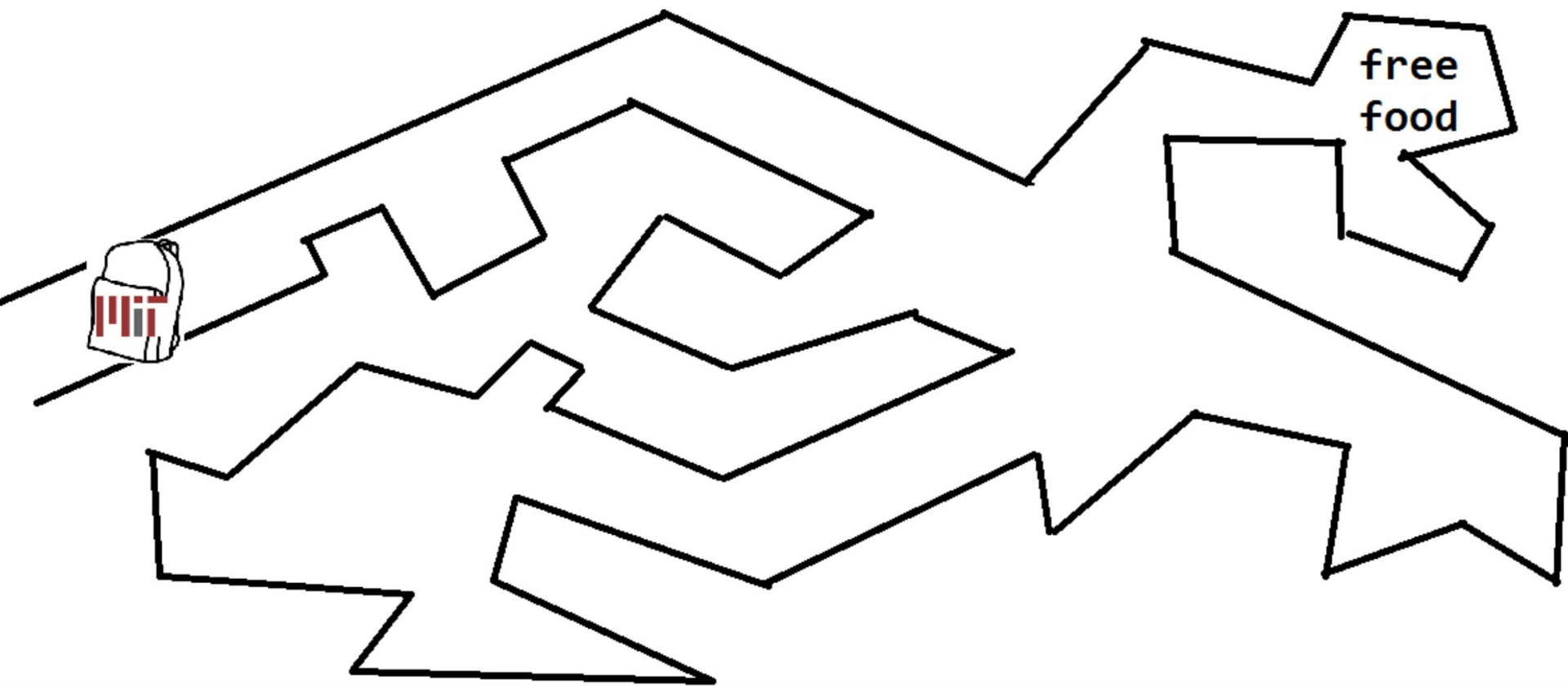
如果右挡住，
直走



如果右侧和
前方被挡住，
向左走



如果右、前、左
受阻，
回去



控制流 - 分支

如果<条件>:

<表达式> <表达式>

...

如果<条件>:

<表达式> <表达式>

...

别的:

<表达式> <表达式>

...

如果<条件>:

<表达式> <表达式>

...

elif <条件>: <表达式> <表达式>

...

别的:

<表达式> <表达式>

...

<condition> 的值为 True 或 False

如果 <condition> 为 True,则评估该块中的表达式

缩进

在 Python 中很重要你如何表示代

码块

```
x = float(input( Enter a number for x: )) y =
```

```
float(input( Enter a number for y: )) if x == y: print( x 和 y 相等 ) print( x
和 y 相等 ) if y != 0: if y != 0: print( 因此,x / y 是 , x/y) print( 因此,x / y 是 , x/
y) elif x < y:
```



```
print( x 更小 ) print( x 更小 )
```

别的:

```
print( y 更小 ) print( y 更小 )
```

= VS ==

```
x = float(input( 输入 x 的数字:  ))
```

```
y = float(input( 请为 y 输入一个数字:  ))
```

```
如果 x == y:
```

```
    print( x 和 y 相等 ) 如果 y != 0:
```

```
        print( 因此,x / y 是 , x/y)
```

```
elif x < y:
```

```
    print( x 更小 )
```

```
别的:
```

```
    print( y 更小 ) print( 谢谢! )
```

What if x = y here?
get a SyntaxError



图片由任天堂提供,保留所有权利。此内容不包含在我们的知识共享许可中。有关详细信息,请参阅<http://ocw.mit.edu/help/faq-fair-use/>。

塞尔达传说 – 迷失森林

继续往前走,带你回到同
一个屏幕,卡在一个循环中

if <exit right>: <set background

```
to exit right: if <exit right>: set background to woods_background
<将背景设置为 woods_background> 将背景设置为 exit_right: if <exit right>
```

依此类推…… 依此类推……

否则:否

<设置背景为 exit_background> <设置背景为 exit_background>

否则:否

<设置背景为 exit_background> <设置背景为 exit_background>

别的:

<设置背景为 exit_background> <设置背景为 exit_background>



塞尔达传说 – 迷失森林

继续往前走,带你回到同
一个屏幕,卡在一个循环中

词云版权未知,保留所有权利。此内容不包含在我们的广告素材中
公共许可。有关详细信息,请参阅<http://ocw.mit.edu/help/faq-fair-use/>。

而<退出右>:

<将背景设置为 woods_background>

<将背景设置为 exit_background>

控制流： while循环

而<条件>：

<表达式>

<表达式>

...

<condition> 计算为布尔值

如果 <condition> 为 True,则执行 while 代码块内的所有步骤

再次检查<条件>

重复直到 <condition> 为 False

while 循环示例

你在失落的森林里。

程序：

```
n = input( 你在失落的森林里。向左还是向右? ) while n == right : n = input( 你在失落的森林里。向左还是向右? )
```

```
print( 你走出了迷失的森林! )
```

控制流： while 和 for 循环

遍历序列中的数字

更复杂的 while 循环 $n = 0$

当 $n < 5$ 时：

打印(n) $n = n + 1$

for 循环的快捷方式 for n in range(5): print(n)

控制流:用于循环

```
for <variable> in range(<some_num>): <expression>  
    <expression>
```

...

每次通过循环,<variable> 取一个值

第一次,<variable> 从最小值开始

下一次,<variable> 获取 prev 值 + 1

等等。

范围（开始、停止、步进）

默认值是 $\text{start} = 0$ 和 $\text{step} = 1$ 和可选循环直到值为 $\text{stop} - 1$

```
mysum = 0  
for  
i in range(7, 10): mysum += i  
    print(mysum)
```

```
mysum = 0  
for  
i in range(5, 11, 2): mysum += i  
    print(mysum)
```

休息声明

立即退出它所在的任何循环跳过代码块中的
剩余表达式

只退出最里面的循环！

而<条件_1>:

而<条件_2>: <表达式_a>

休息

<表达式_b>

<表达式_c>

休息声明

我的总和 = 0

对于范围内的 i (5, 11, 2):

我的和 += 我 我的和

如果 mysumb == 5: 如果

休息休息

我的总和 += 1

打印（我的总和）

这个程序会发生什么？

为了

VS while 循环

for 循环

while 循环

可以使用while循环**重写**for
循环

结束通过休息提前

循环内提前结束

麻省理工学院开放课件[https://
ocw.mit.edu](https://ocw.mit.edu)

6.0001 计算机科学和 Python 编程简介
2016 年秋季

有关引用这些材料或我们的使用条款的信息,请访问: <https://ocw.mit.edu/terms>。