



# 从 Blockly 到 Python 数据分析与可视化

上海师范大学-李鲁群教授实验室

李鲁群（教授）

[success@shnu.edu.cn](mailto:success@shnu.edu.cn)



上海师范大学  
SHANGHAI NORMAL UNIVERSITY



# 简介:

本章主要介绍什么是Blockly、Blockly编程环境、Blockly模块功能，Blockly与Python教学环境搭建，并开发一个“Hello world”程序。

本章的学习目标:

- 了解Blockly的概况;
- 了解Blockly编程环境;
- 熟悉Blockly模块功能的使用;
- 掌握Blockly与Python教学环境的搭建;
- 掌握利用Blockly进行程序开发。



# 本章内容

授课内容：

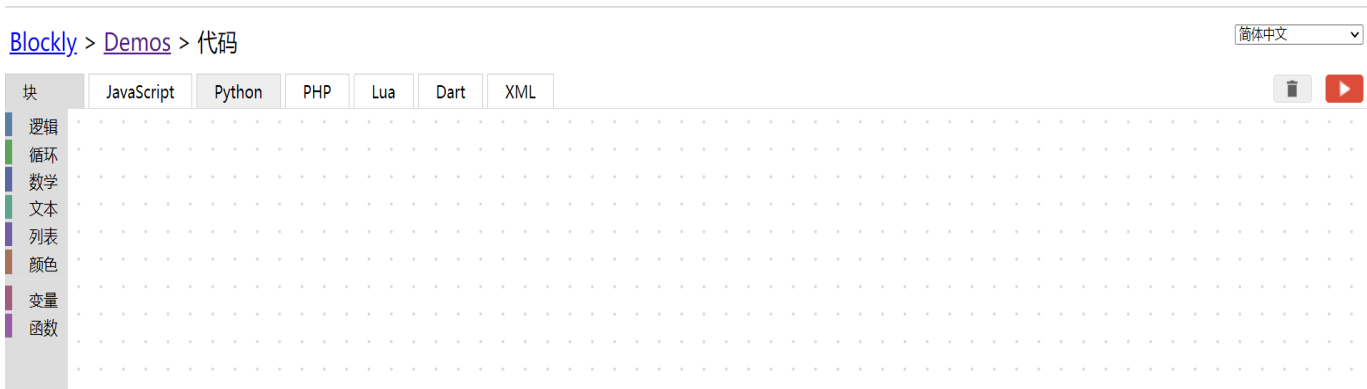
1. Blockly概述
2. Blockly编程环境
3. Blockly模块功能
4. Blockly与Python教学环境搭建
5. 开发“Hello world”程序
6. 本章小结
7. 练习



# 1.1 Blockly概述

## 1.1.1 什么是Blockly

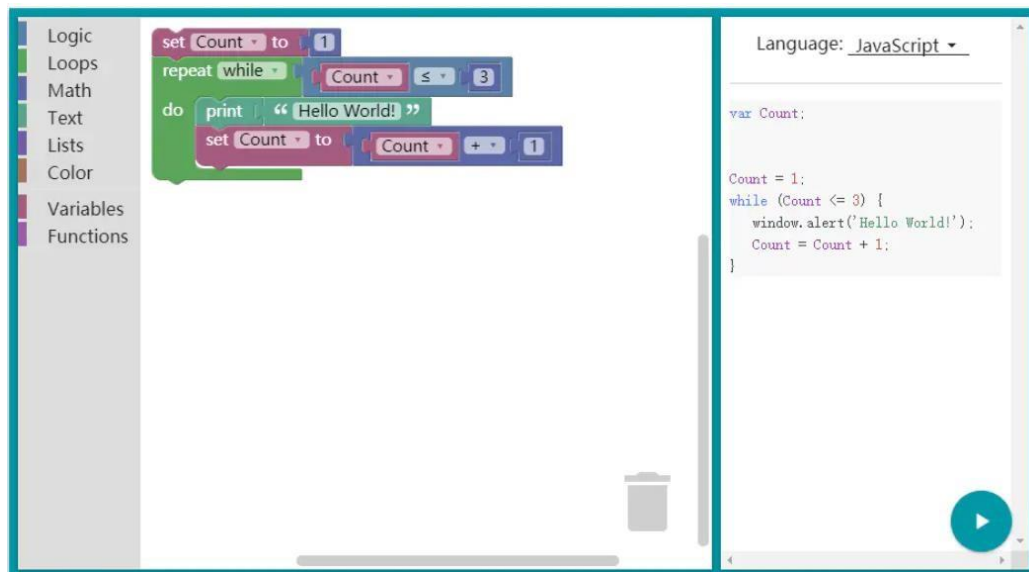
- 2012年6月，Google发布了基于网页的完全可视化的编程语言Google Blockly。
- 每个图形对象都是代码块，你可以将它们拼接起来，创造出简单功能，然后将一个个简单功能组合起来，构建一个程序。
- Blockly可以在线开发，也可以离线开发，其代码可以转换成多种环境语言代码：例如JavaScript、Python、PHP、Lua、Dart



# 1.1 关于Blockly

## 1.1.1 什么是Blockly

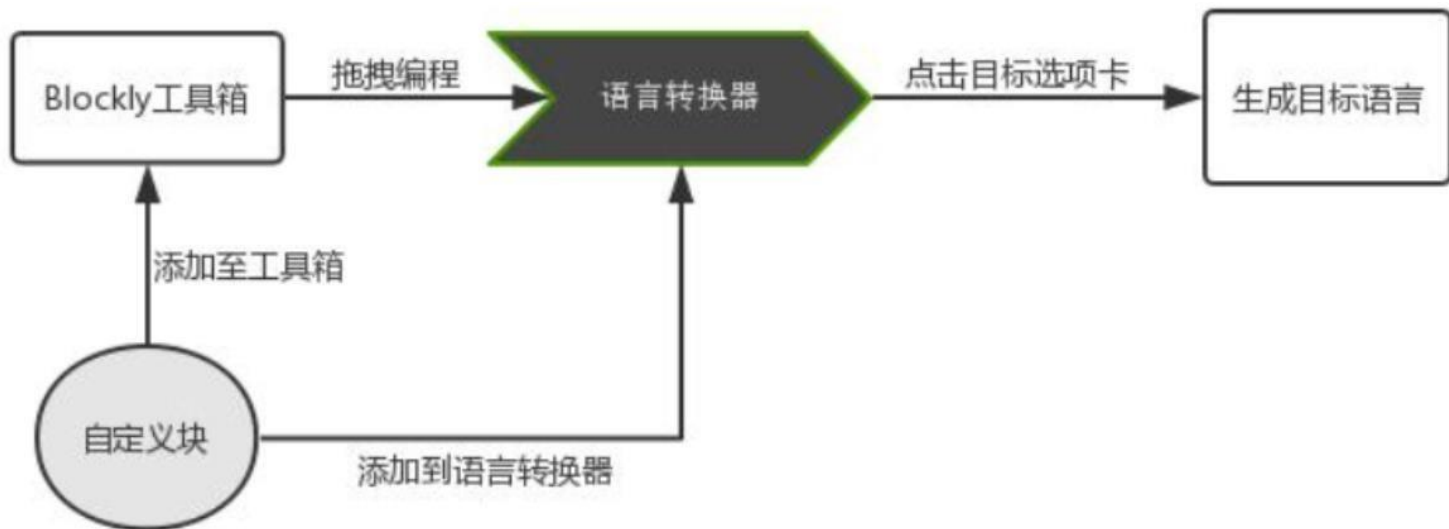
Blockly提供了强大的自定义模块功能，所以，理论上它能把任何基于文本的逻辑（程序或配置文件）可视化。在需要编辑复杂的逻辑或大量灵活配置的地方，都可以用Blockly改善用户体验，减少错误概率，降低学习成本。例如，游戏逻辑编辑器、艺术装置的控制、数字图像和动画程序化生成等等。下图展示了Blockly的核心功能：将图形化的积木块逻辑（左侧）转化为代码。



# 1.1 Blockly概述

## 1.1.1 什么是Blockly

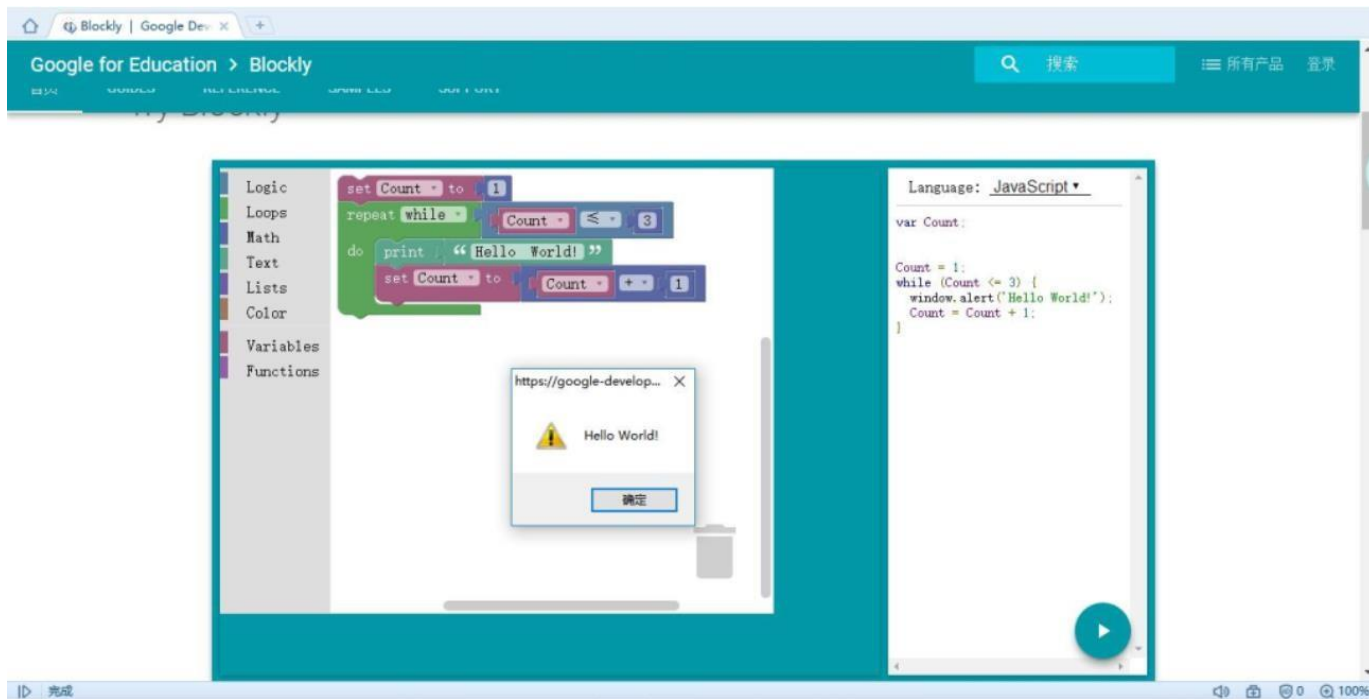
- 我们可以使用Blockly的工具箱编写Blockly代码块实现程序功能，然后使用语言转换器转换为需要的语言。
- 我们也可以自定义Blockly代码块生成工具放入工具箱和语言转换器中，用于以后使用。



# 1.2 Blockly编程环境

## 1.2.1 Blockly的两种编程环境

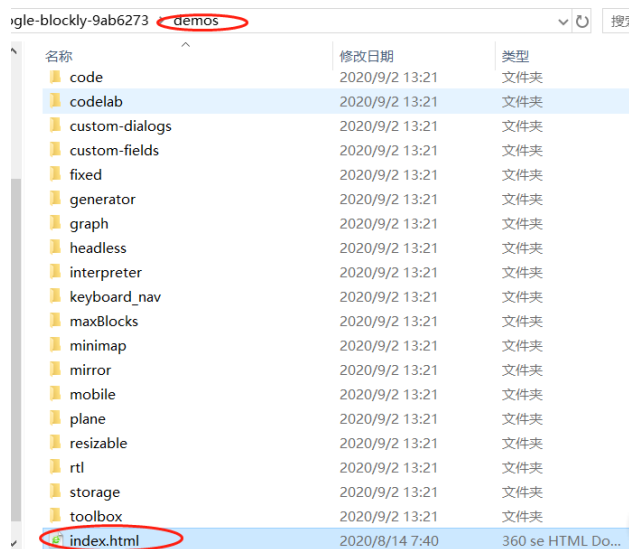
- Blockly在线编程环境: <https://developers.google.cn/blockly/>



# 1.2 Blockly编程环境

## 1.2.1 Blockly的两种编程环境

- Blockly离线编程环境: <https://github.com/google/blockly>



### Blockly > Demos

These demos are intended for developers who want to integrate Blockly with their own applications.



#### Fixed Blockly

Inject Blockly into a page as a fixed element.



#### Resizable Blockly

Inject Blockly into a page as a resizable element.



#### Defining the Toolbox

Organize blocks into categories for the user.



#### Maximum Block Limit

Limit the total number of blocks allowed (for academic exercises).



Math.sin(0)

#### Generate JavaScript

Turn blocks into code and execute it.



# 1.2 Blockly编程环境

## 1.2.1 Blockly的两种编程环境

- Blockly界面



# 1.2 Blockly编程环境

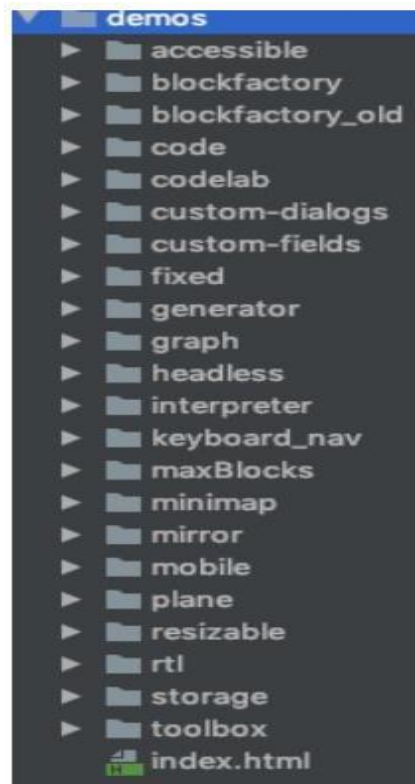
## 1.2.1 Blockly的两种编程环境

- Blockly界面
- 分类区 (Categories)：用于将Blocks按模块进行分类，Blockly默认定义了8种分类，每种分类使用不同的颜色，有不同功能，譬如Loops分类存放循环积木，Math分类存放数学操作积木等；
- 积木区 (Blocks)：用于存放分类下的积木，不同积木代表不同的功能，在Blockly中，积木可以使用开发者工具，或者编写代码灵活定义积木；
- 脚本区 (Scripts)：用于编写代码的区域，将积木区中的积木，拖动到脚本区，积木进行卡合形成积木组，积木组即可成为可执行的代码段，执行代码段获得运算结果；
- 工作空间 (Workspace)：分类区、积木区、脚本区组合为工作空间，Blockly操作以Workspace为核心。

# 1.2 Blockly编程环境

## 1.2.1 Blockly的两种编程环境

- Blockly的一些demo例子



# 1.2 Blockly编程环境

## 1.2.1 Blockly的两种编程环境

- 离线Blockly的一些demo例子

- **Fixed Blockly** : 固定大小的Blockly块的工作空间，不会随浏览器大小变化而变化。
- **Resizable Blockly** : 可变大小的Blockly块的工作空间，会随浏览器大小变化而变化。
- **Defining the Toolbox** : Blockly自定义的工具箱
- **Maximum Block Limit** : 限制Blockly块的个数，不能超过指定Blockly个数。
- **Generate JavaScript** : 生成JavaScript代码
- **Headless** : 将xml代码生成其他的语言的代码
- **JS Interpreter** : 展示使用Blockly代码块生成JavaScript代码并可以单步执行。
- **Graph** : 通过数学公式在坐标系中生成图形
- **RTL** : 从右到左显示Blockly块（对于阿拉伯语和希伯来语使用）

Blockly > Demos

These demos are intended for developers who want to integrate Blockly with their own applications.



**Fixed Blockly**  
Inject Blockly into a page as a fixed element.



**Resizable Blockly**  
Inject Blockly into a page as a resizable element.



**Defining the Toolbox**  
Organize blocks into categories for the user.



**Maximum Block Limit**  
Limit the total number of blocks allowed (for academic exercises).



**Generate JavaScript**  
Turn blocks into code and execute it.

# 1.2 Blockly编程环境

## 1.2.1 Blockly的两种编程环境

- 离线Blockly的一些demo例子（续）

- Custom Dialogs : 自定义实现覆盖浏览器对话框
- Custom Fields : 自定义字段
- Cloud Storage : 云存储在App Engine
- Mirrored Blockly : 展示两个同步的Blockly工作区, 左侧主工作区变化, 右侧从工作区也发生变化。
- Accessible Blockly : 屏幕阅读无障碍访问版本
- Plane : 支持35种语言
- Code Editor : 将Blockly程序导出成JavaScript,Python,PHP,Lua,Dart或XML
- Blockly Developer Tools : 使用Blockly构建自定义的代码块并设置到工作箱。

# 1.2 Blockly编程环境

## 1.2.2 Blockly games:

- 在线: <https://blockly.games/>
- 离线下载: <https://github.com/google/blockly-games/wiki/Offline>

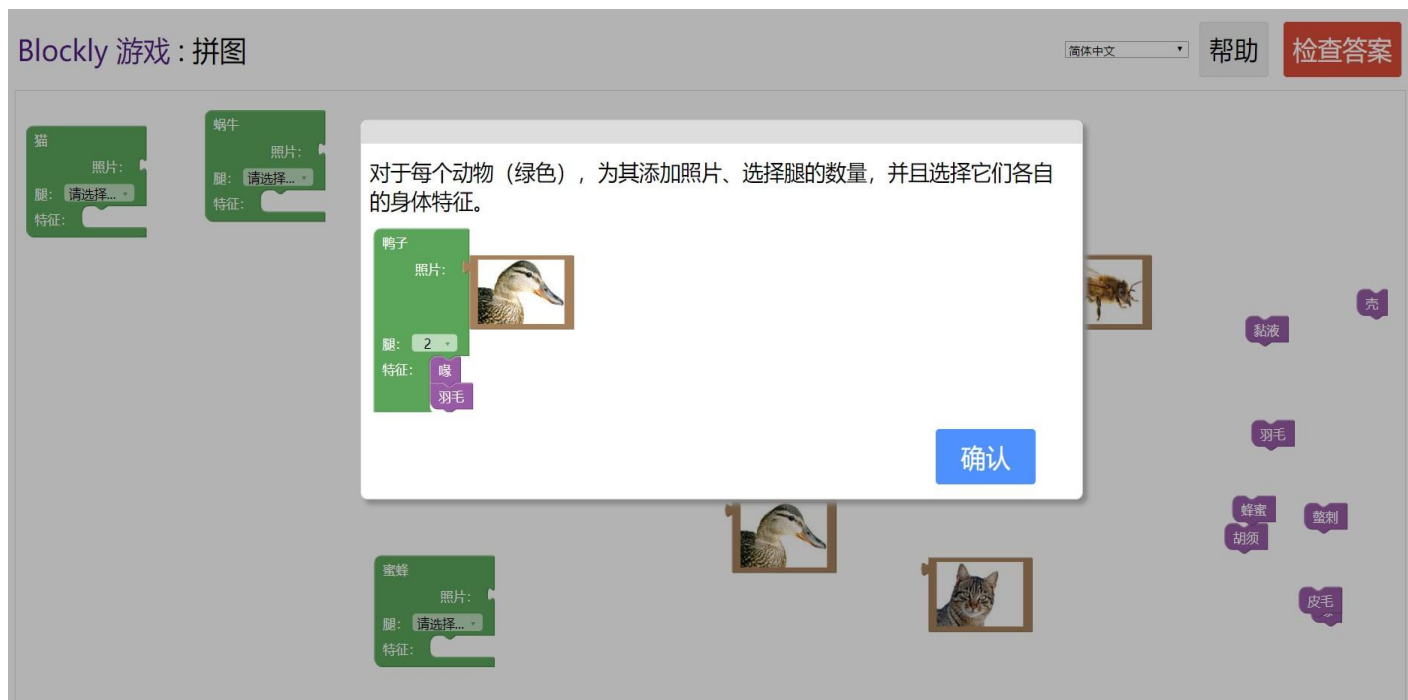
# 1.2 Blockly编程环境

- Blockly Games



# 1.2 Blockly编程环境

- 拼图游戏





# 1.2 Blockly编程环境

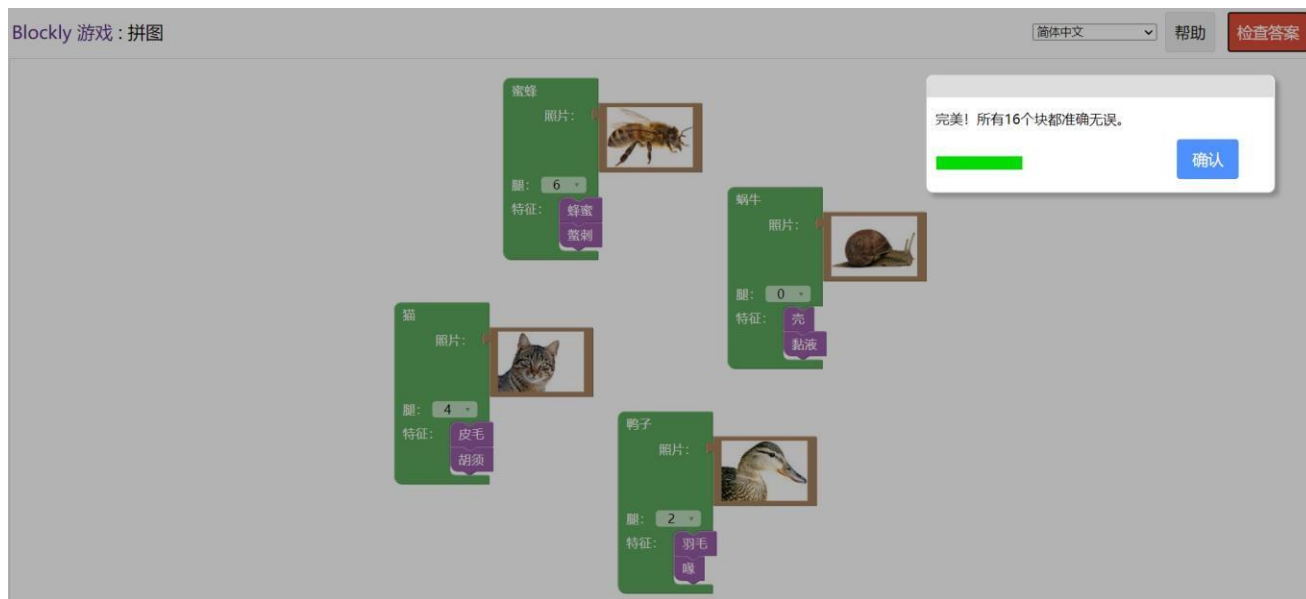
- 体验一下Blockly Games的拼图游戏

Blockly 游戏：拼图



# 1.2 Blockly编程环境

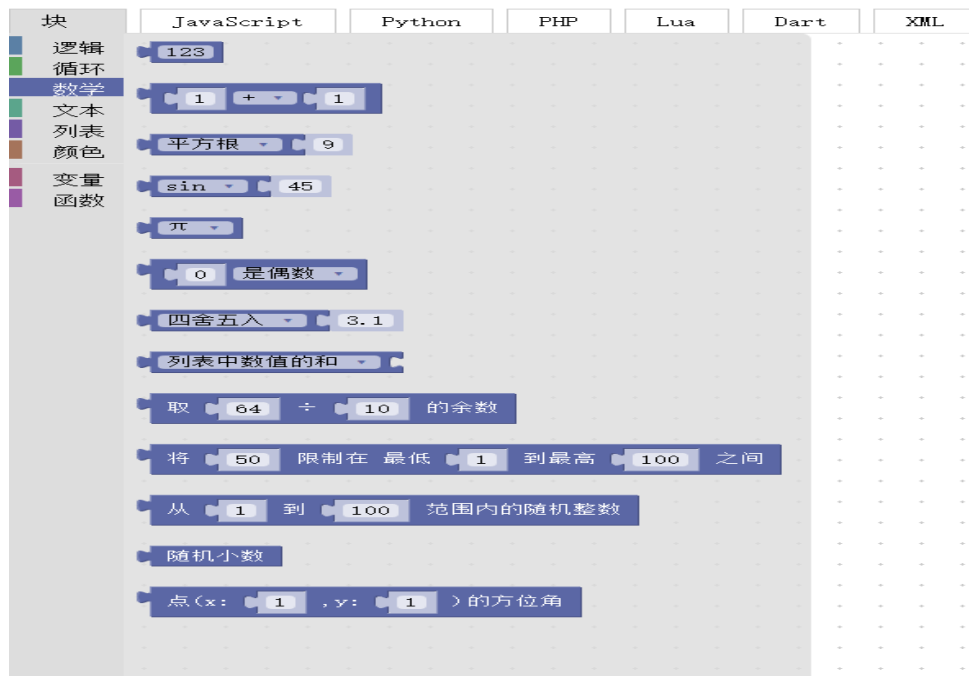
- 体验一下Blockly Games的拼图游戏



# 1.3 Blockly模块功能

- 工具箱中有8个主要模块：
- `blockly-master>demos>code>index.html` 中可以查看源码
- 运行`index.html`结果（右上角可以修改语言类型为简体中文）

[Blockly](#) > [Demos](#) > 代码

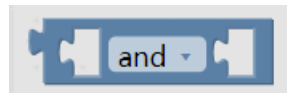


# 1.3 Blockly模块功能

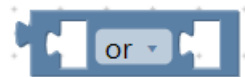
## 1.3.1 逻辑模块

- 逻辑运算符有3种：

(1) &&     逻辑与运算符



(2) ||     逻辑或运算符



(3) !     逻辑非运算符



# 1.3 Blockly模块功能

## 1.3.1 逻辑模块

- 条件逻辑语句

格式：如果xx，则执行xx



# 1.3 Blockly模块功能

## 1.3.1 逻辑模块

- 不等（相等）关系语句

格式：左边 = 右边  
左边  $\neq$  右边  
左边  $<$  右边  
左边  $\leq$  右边  
左边  $\geq$  右边



# 1.3 Blockly模块功能

## 1.3.1 逻辑模块

- 真假关系语句

格式：真 True  
假 False



# 1.3 Blockly模块功能

## 1.3.1 逻辑模块

- 断言语句

格式：断言xx  
如果为真，则xx  
如果为假，则xx





# 1.3 Blockly模块功能

## 1.3.2 循环模块

- 简单循环语句

格式：重复xx次数  
执行xx

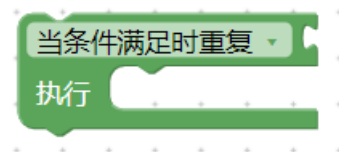


# 1.3 Blockly模块功能

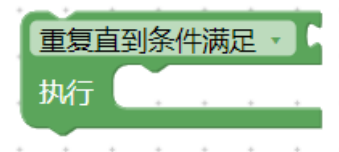
## 1.3.2 循环模块

- 条件循环语句

格式：（1）当满足条件时重复xx  
执行xx



（2）重复直到条件满足xx  
执行xx



# 1.3 Blockly模块功能

## 1.3.2 循环模块

- 利用变量控制循环

格式：变量xx 从 xx 数到 xx ， 每次增加xx  
执行xx

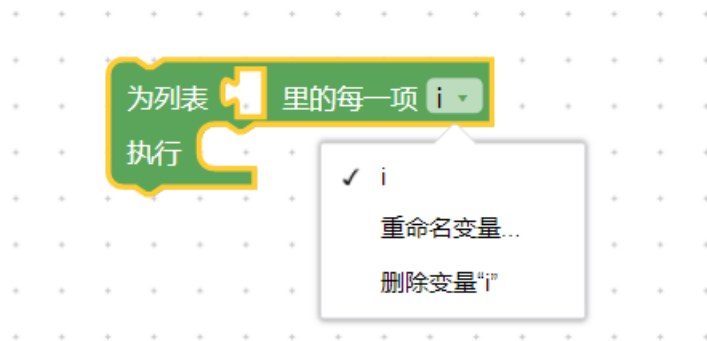


# 1.3 Blockly模块功能

## 1.3.2 循环模块

- 对列表元素进行循环操作

格式：为列表xx 里的每一项xx  
执行xx

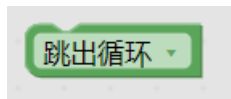


# 1.3 Blockly模块功能

## 1.3.2 循环模块

- 结束 / 继续循环操作

格式：跳出循环



继续下一轮循环



# 1.3 Blockly模块功能

## 1.3.3 数学模块

- 简单数字

格式：xx（数字）

例如：1，10，100，999.....



# 1.3 Blockly模块功能

## 1.3.3 数学模块

- 简单数字计算

格式: **xx** (数字) + **xx** (数字)

**xx** (数字) - **xx** (数字)

**xx** (数字) × **xx** (数字)

**xx** (数字) ÷ **xx** (数字)

**xx** (数字) ^ **xx** (数字)



# 1.3 Blockly模块功能

## 1.3.3 数学模块

- 简单数字计算

格式：平方根 xx (数字)

绝对值 xx (数字)

负号 xx (数字)

ln xx (数字)

log 10 xx (数字)

e^ xx (数字)

10^ xx (数字)



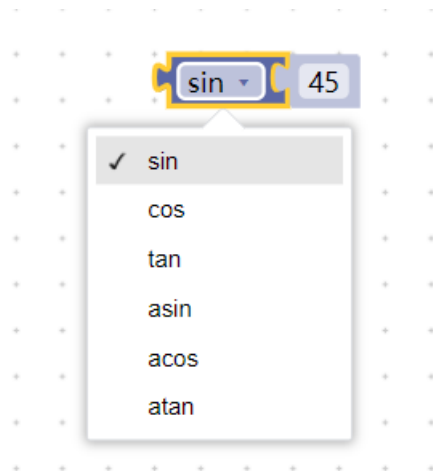


# 1.3 Blockly模块功能

## 1.3.3 数学模块

- 三角函数值计算

格式:  $\sin$  xx (数字)  
 $\cos$  xx (数字)  
 $\tan$  xx (数字)  
 $\text{asin}$  xx (数字)  
 $\text{acos}$  xx (数字)  
 $\text{atan}$  xx (数字)



# 1.3 Blockly模块功能

## 1.3.3 数学模块

- 常见常量运算

格式:  $\pi$

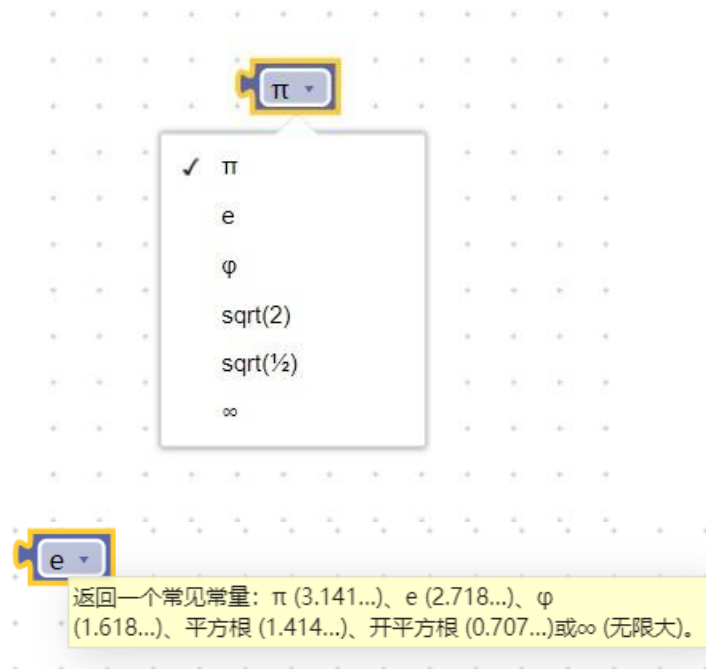
$e$

$\varphi$

$\text{sqrt}(2)$

$\text{sqrt}(1/2)$

$\infty$



# 1.3 Blockly模块功能

## 1.3.3 数学模块

- 简单数字判断

格式：xx 是 偶数

xx 是 奇数

xx 是 质数

xx 是 整数

xx 是 正数

xx 是 负数

xx 可被整除



# 1.3 Blockly模块功能

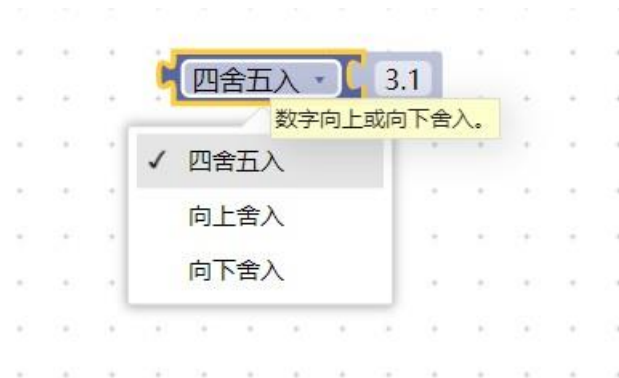
## 1.3.3 数学模块

- 四舍五入

格式：四舍五入 xx（数字）

向上舍入 xx（数字）

向下舍入 xx（数字）



# 1.3 Blockly模块功能

## 1.3.3 数学模块

- 对列表元素进行运算

格式：列表中数值的和

列表最小值

列表最大值

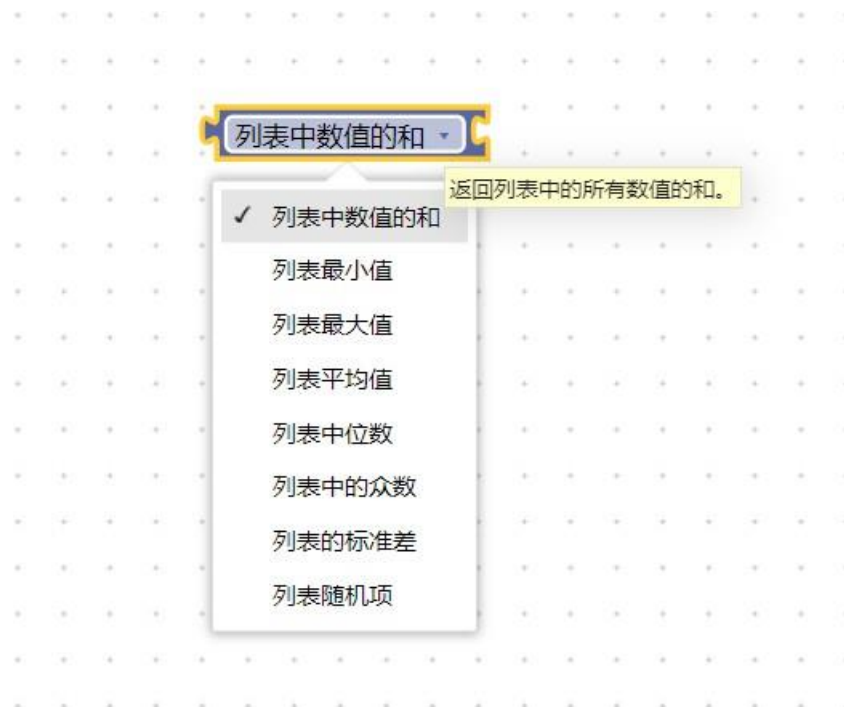
列表平均值

列表中位数

列表的众数

列表的标准差

列表的随机项



# 1.3 Blockly模块功能

## 1.3.3 数学模块

- 取余数运算

格式：取 $xx \div xx$  的余数



# 1.3 Blockly模块功能

## 1.3.3 数学模块

- 限制运算

格式：将xx 限制在 最低xx 到最高xx 之间



# 1.3 Blockly模块功能

## 1.3.3 数学模块

- 随机整数

格式：从xx 到xx 范围内的随机整数





# 1.3 Blockly模块功能

## 1.3.3 数学模块

- 随机小数

格式：随机小数



# 1.3 Blockly模块功能

## 1.3.3 数学模块

- 方位角

格式：点（x: ， y: ）的方位角



# 1.3 Blockly模块功能

## 1.3.4 文本模块

- 简单文本  
格式：“ ”



# 1.3 Blockly模块功能

## 1.3.4 文本模块

- 通过串起任意数量的项以建立一段文本  
格式：建立文本从xxxx



# 1.3 Blockly模块功能

## 1.3.4 文本模块

- 将一些文本追加到“变量”项目里  
格式：向 项目xx 附加文本 “ ”

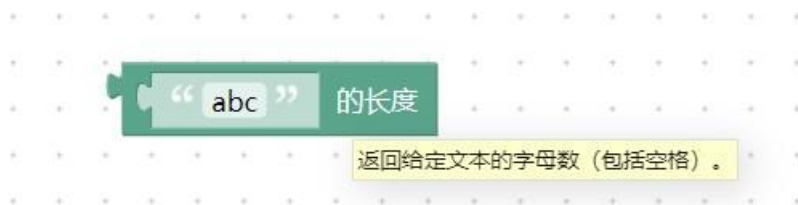


# 1.3 Blockly模块功能

## 1.3.4 文本模块

- 文本长度

格式：“ ” 的长度



# 1.3 Blockly模块功能

## 1.3.4 文本模块

- 文本为空  
如果给定的文本为空，则返回真  
格式：“ ” 是空的



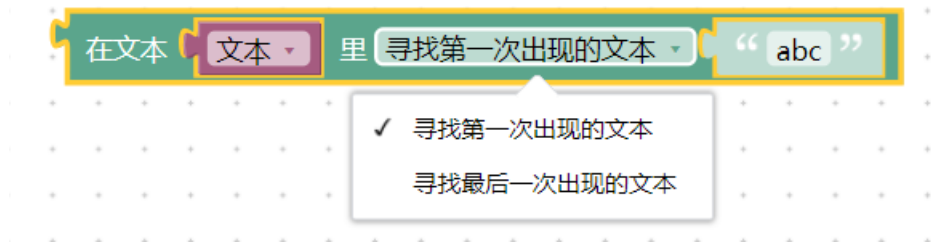
# 1.3 Blockly模块功能

## 1.3.4 文本模块

- 寻找文本

返回第一个文本段在第二个文本段里的第一个/最后一个匹配项的起始位置，如果没有，则返回0。

格式：在文本 文本1 里寻找第一次/最后一次出现的文本 “ ”





# 1.3 Blockly模块功能

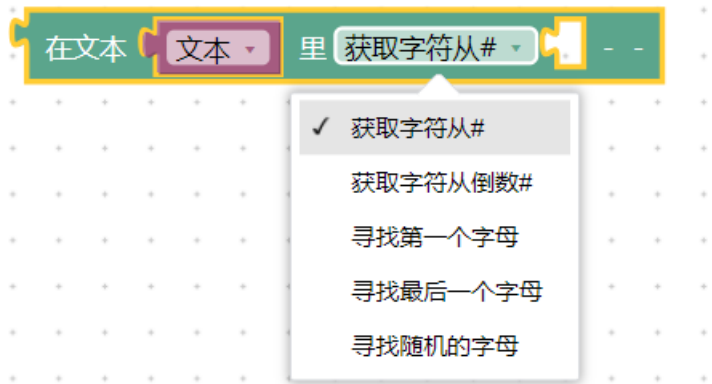
## 1.3.4 文本模块

- 获取文本

返回位于指定位置的字母，#1为第一项。

格式：在文本 文本1 里获取字符从# / 获取字符从倒数# xxxx

在文本 文本1 里寻找第一个字母 / 最后一个字母 / 随机的字母



# 1.3 Blockly模块功能

## 1.3.4 文本模块

- 返回文本

返回文本中指定的一部分。

格式：从文本 文本1 取得子字符串自# xxxx/ 子字符串自倒数# xxxx / 子字符串自第一个字符 至 字符# xxxx / 到最后一个字符# xxxx / 到最后一个字符



# 1.3 Blockly模块功能

## 1.3.4 文本模块

- 转换文本

用不同的大小写模式复制并返回这段文本。

格式：转为大写/小写/首字母大写 “ ”



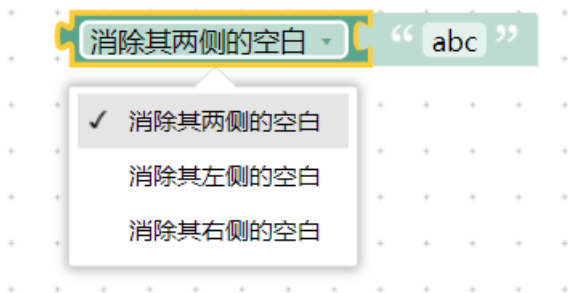
# 1.3 Blockly模块功能

## 1.3.4 文本模块

- 删除文本中的空白

从某一端或者从两端同时删除多余的空白，并返回这段文本的一个副本。

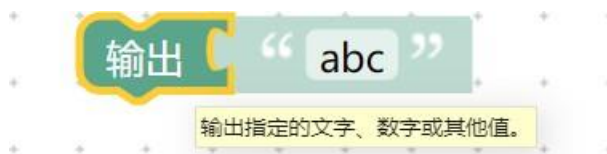
格式：消除其两侧 / 左侧 / 右侧的空白 “ ”



# 1.3 Blockly模块功能

## 1.3.4 文本模块

- 输出文本  
输出指定的文字、数字或其他值。  
格式：输出 “ ”



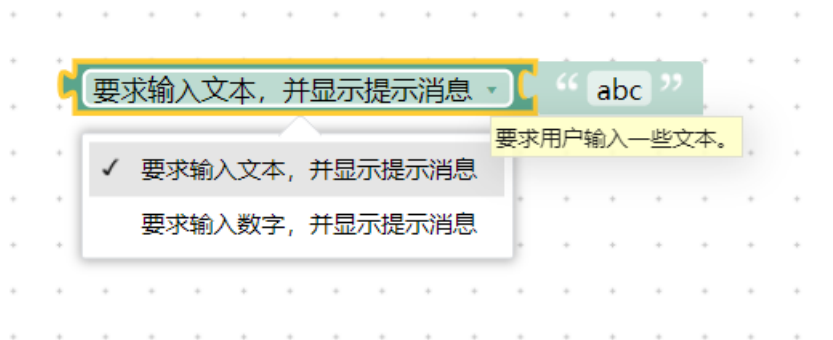
# 1.3 Blockly模块功能

## 1.3.4 文本模块

- 输入文本

要求用户输入一些文本。

格式：要求输入文本，并显示提示消息 “ ”



# 1.3 Blockly模块功能

## 1.3.5 列表模块

- 创建列表  
建立一个具有任意数量项目的列表。  
格式：创建空列表



# 1.3 Blockly模块功能

## 1.3.5 列表模块

- 创建列表  
建立一个具有任意数量项目的列表。  
格式：创建列表从





# 1.3 Blockly模块功能

## 1.3.5 列表模块

- 重复创建列表  
建立包含指定重复次数的值的列表。  
格式：建立列表使用项xxxx 重复xx



# 1.3 Blockly模块功能

## 1.3.5 列表模块

- 列表长度  
返回列表的长度。  
格式：xxxx的长度



# 1.3 Blockly模块功能

## 1.3.5 列表模块

- 列表判定为空  
如果该列表为空，则返回真。  
格式：xxxx 是 空的



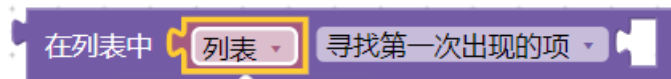
# 1.3 Blockly模块功能

## 1.3.5 列表模块

- 查找列表项

返回在列表中的第一个/最后一个匹配项的索引值，如果找不到则返回0。

格式：xxxx 是 空的



✓ 列表  
重命名变量...  
删除变量“列表”



✓ 寻找第一次出现的项  
寻找最后一次出现的项

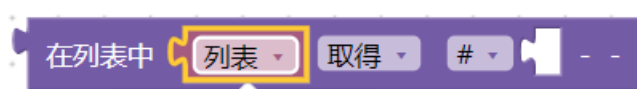
# 1.3 Blockly模块功能

## 1.3.5 列表模块

- 返回列表项

返回在列表中的指定位置的项，#1是第一项。

格式：如下图



- ✓ 列表
- 重命名变量...
- 删除变量“列表”



- ✓ #
- 倒数第#
- 第一个
- 最后一个
- 随机的



- ✓ 取得
- 取得并移除
- 移除

# 1.3 Blockly模块功能

## 1.3.5 列表模块

- 设置列表项

设置在列表中指定位置的项，#1是第一项。

格式：如下图



# 1.3 Blockly模块功能

## 1.3.5 列表模块

- 复制列表  
复制列表中指定的部分。  
格式：如下图



# 1.3 Blockly模块功能

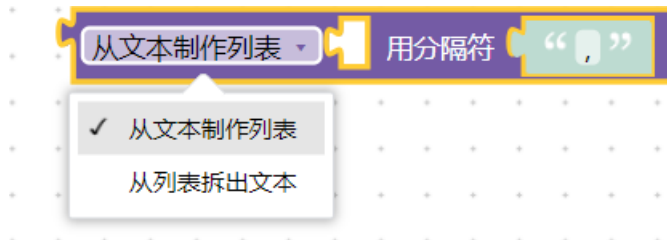
## 1.3.5 列表模块

- 文本拆分与分隔

1 拆分文本到文本列表，按每个分隔符拆分。

2加入文本列表至一个文本，由分隔符分隔。 格

式：如下图

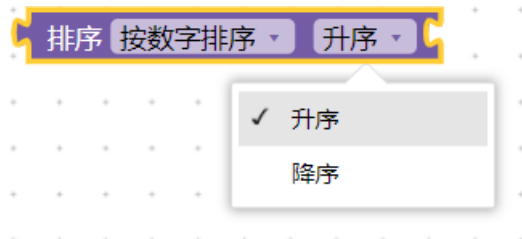
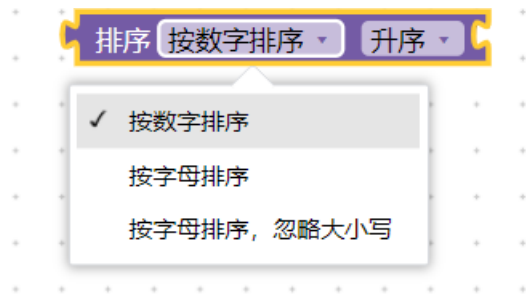




# 1.3 Blockly模块功能

## 1.3.5 列表模块

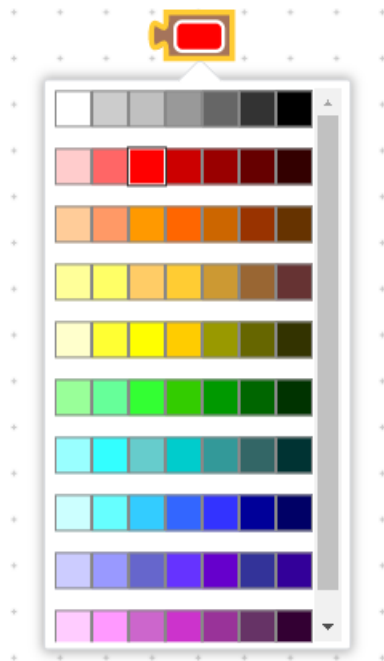
- 排序  
排序一个列表的拷贝。  
格式：如下图



# 1.3 Blockly模块功能

## 1.3.6 颜色模块

- 选择颜色  
从调色板中选择一个颜色。  
格式：如下图



# 1.3 Blockly模块功能

## 1.3.6 颜色模块

- 随机颜色  
随机选择一种颜色。  
格式：如下图



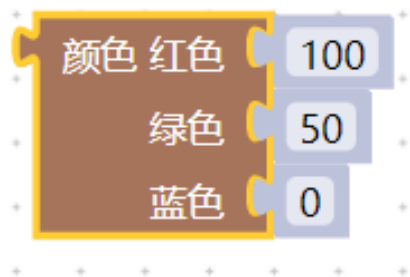
# 1.3 Blockly模块功能

## 1.3.6 颜色模块

- 创建颜色

通过指定红色、绿色、蓝色的量创建一种颜色，所有的值必须在0-100之间。

格式：如下图



# 1.3 Blockly模块功能

## 1.3.6 颜色模块

- 混合颜色

把两种颜色以混合比例（0.0-1.0）进行混合。

格式：如下图

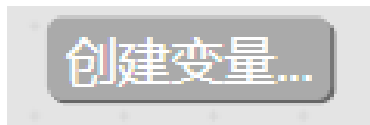


# 1.3 Blockly模块功能

## 1.3.7 变量模块

- 创建变量

格式：如下图



# 1.3 Blockly模块功能

## 1.3.7 变量模块

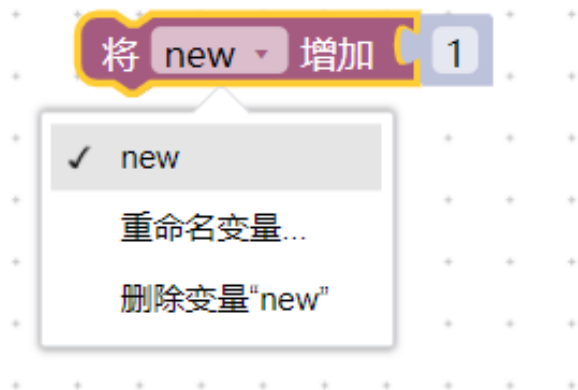
- 变量赋值  
设置此变量，以使它和输入值相等。  
格式：如下图



# 1.3 Blockly模块功能

## 1.3.7 变量模块

- 变量增值  
为返回此变量的值。  
格式：如下图





# 1.3 Blockly模块功能

## 1.3.7 变量模块

- 返回变量  
为变量增加一个数值。  
格式：如下图



# 1.3 Blockly模块功能

## 1.3.8 函数模块

- 创建函数  
创建一个不带输出值的函数。  
格式：如下图



# 1.3 Blockly模块功能

## 1.3.8 函数模块

- 创建函数  
创建一个带输出值的函数。  
格式：如下图



# 1.3 Blockly模块功能

## 1.3.8 函数模块

- 函数  
如果值为真，则返回第二个值。  
格式：如下图



# 1.4 Python简介

- 什么是Python?
- Python是一款易于学习且功能强大的开放源代码的编程语言，它可以快速帮助人们完成各种编程任务，并且能够把其他语言制作的各種模块很轻松地联结在一起。使用Python编写的程序可以在绝大多数的平台上顺利运行。

特点	含义
简单、易学	有极其简单的说明文档
丰富的库	标准库庞大、功能齐全
可扩展、可嵌入	可以把Python嵌入C、C++程序
面向对象、高层	程序是由数据和功能组合而成的对象构建起来的
解释性	程序不需要编译成二进制代码，可以直接从源代码运行程序
免费开源、可移植	可以阅读、拷贝、改动源代码

# 1.4 Python简介

## 1.4.1 Python安装

- 软件下载: <https://www.python.org/downloads/>
- 根据你的操作系统选择下载版本



# 1.4 Python简介

## 1.4.2 Python下载不同版本

Looking for a specific release?

Python releases by version number:

Release version	Release date	Click for more	
<a href="#">Python 3.8.3rc1</a>	April 29, 2020	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 2.7.18</a>	April 20, 2020	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.7.7</a>	March 10, 2020	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.8.2</a>	Feb. 24, 2020	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.8.1</a>	Dec. 18, 2019	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.7.6</a>	Dec. 18, 2019	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.6.10</a>	Dec. 18, 2019	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.5.9</a>	Nov. 2, 2019	<a href="#">Download</a>	<a href="#">Release Notes</a>

[View older releases](#)



# 1.4 Python简介

## 1.4.3 选择Python下载文件

### Python Releases for Windows

- [Latest Python 3 Release - Python 3.8.2](#)
- [Latest Python 2 Release - Python 2.7.18](#)

#### Stable Releases

- [Python 3.8.3rc1 - April 29, 2020](#)

**Note that Python 3.8.3rc1 cannot be used on Windows XP or earlier.**

- [Download Windows help file](#)
- [Download Windows x86-64 embeddable zip file](#)
- [Download Windows x86-64 executable installer](#)
- [Download Windows x86-64 web-based installer](#)
- [Download Windows x86 embeddable zip file](#)
- [Download Windows x86 executable installer](#)
- [Download Windows x86 web-based installer](#)

64位离线安装文件

64位在线安装

32位

- [Python 2.7.18 - April 20, 2020](#)

- [Download Windows debug information files](#)
- [Download Windows debug information files for 64-bit binaries](#)

#### Pre-releases

- [Python 3.9.0a6 - April 28, 2020](#)

- [Download Windows help file](#)
- [Download Windows x86-64 embeddable zip file](#)
- [Download Windows x86-64 executable installer](#)
- [Download Windows x86-64 web-based installer](#)
- [Download Windows x86 embeddable zip file](#)
- [Download Windows x86 executable installer](#)
- [Download Windows x86 web-based installer](#)

- [Python 2.7.18rc1 - April 4, 2020](#)

- [Download Windows debug information files](#)
- [Download Windows debug information files for 64-bit binaries](#)
- [Download Windows help file](#)
- [Download Windows x86-64 MSI installer](#)





# 1.4 Python简介

## 1.4.4 Python安装和使用

- 双击安装，记得勾选红框，不勾选的话安装完成后需要自己配置环境变量。



# 1.4 Python简介

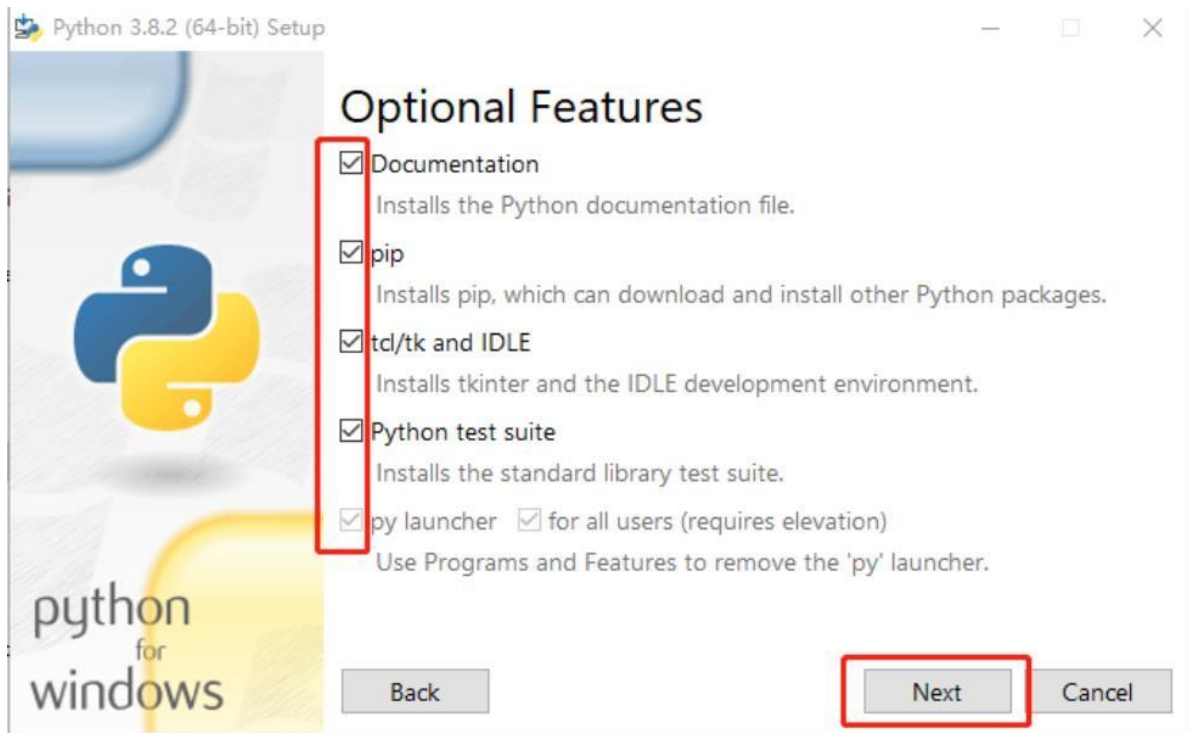
## 1.4.4 Python安装和使用

- 勾选【Add Python 3.8 to PATH】
- 然后点击【Customize installation】



# 1.4 Python简介

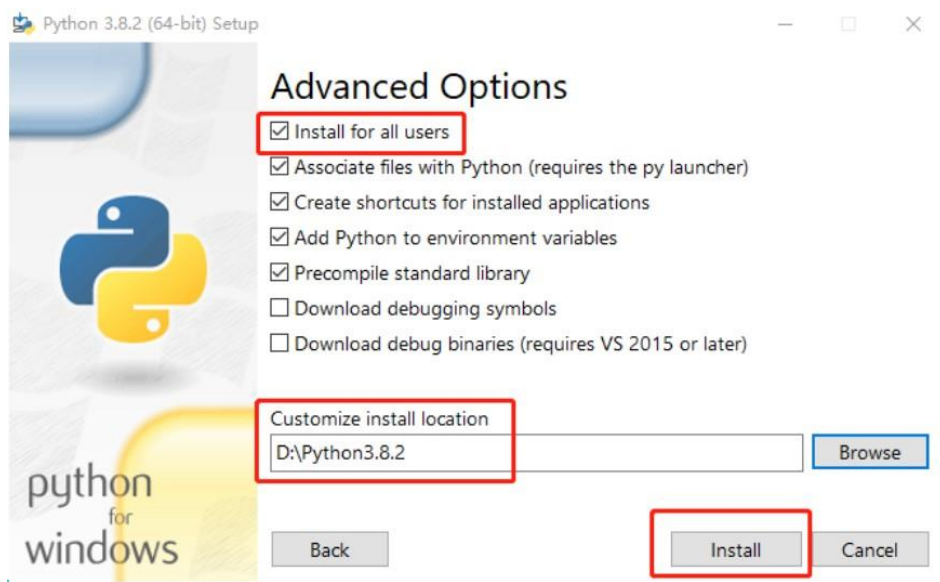
## 1.4.4 Python安装和使用



# 1.4 Python简介

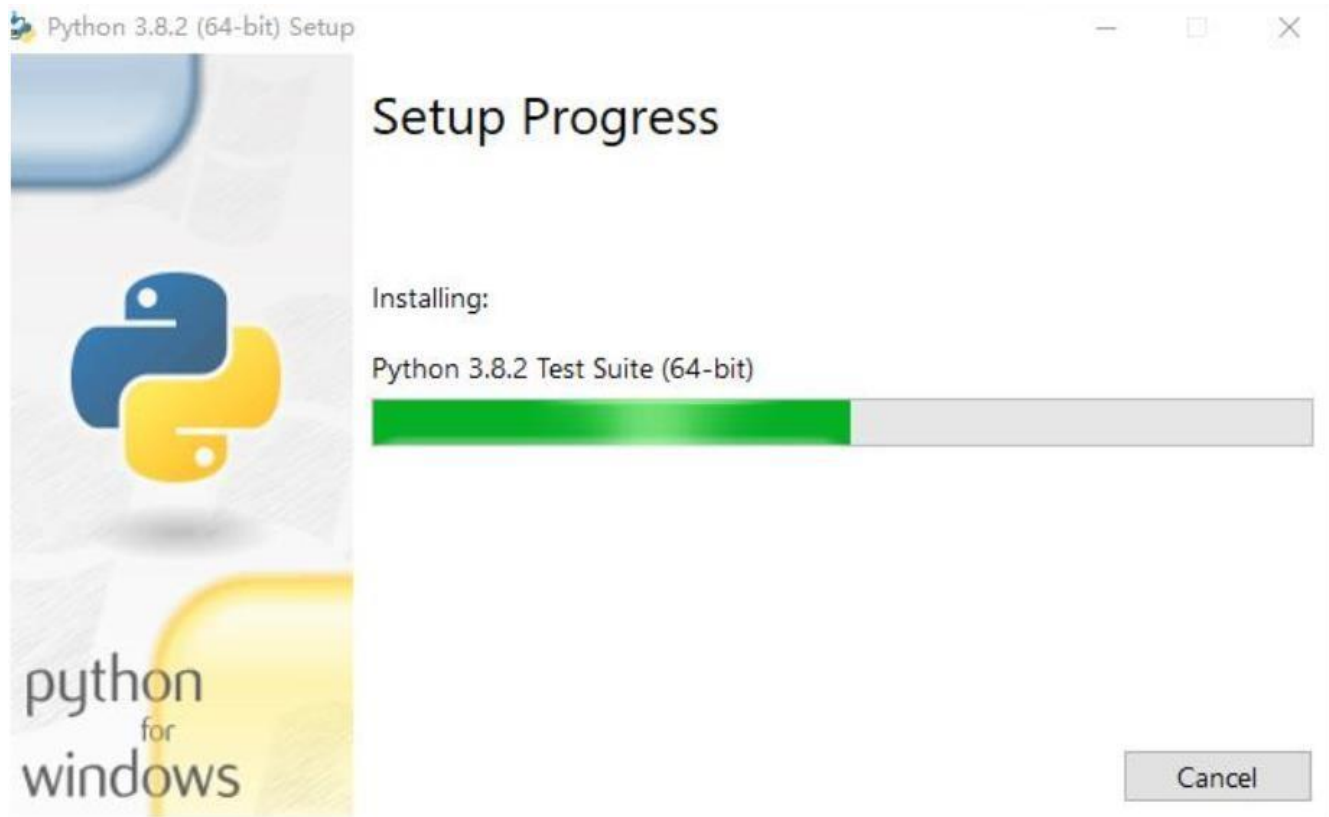
## 1.4.4 Python安装和使用

- 勾选【Install for all users】
- 再点击【Browse】更改软件的安装路径，建议安装在除C盘之外的其他盘
- 可以在D盘或者其他盘新建一个【Python3.8】文件夹，然后点击【Install】



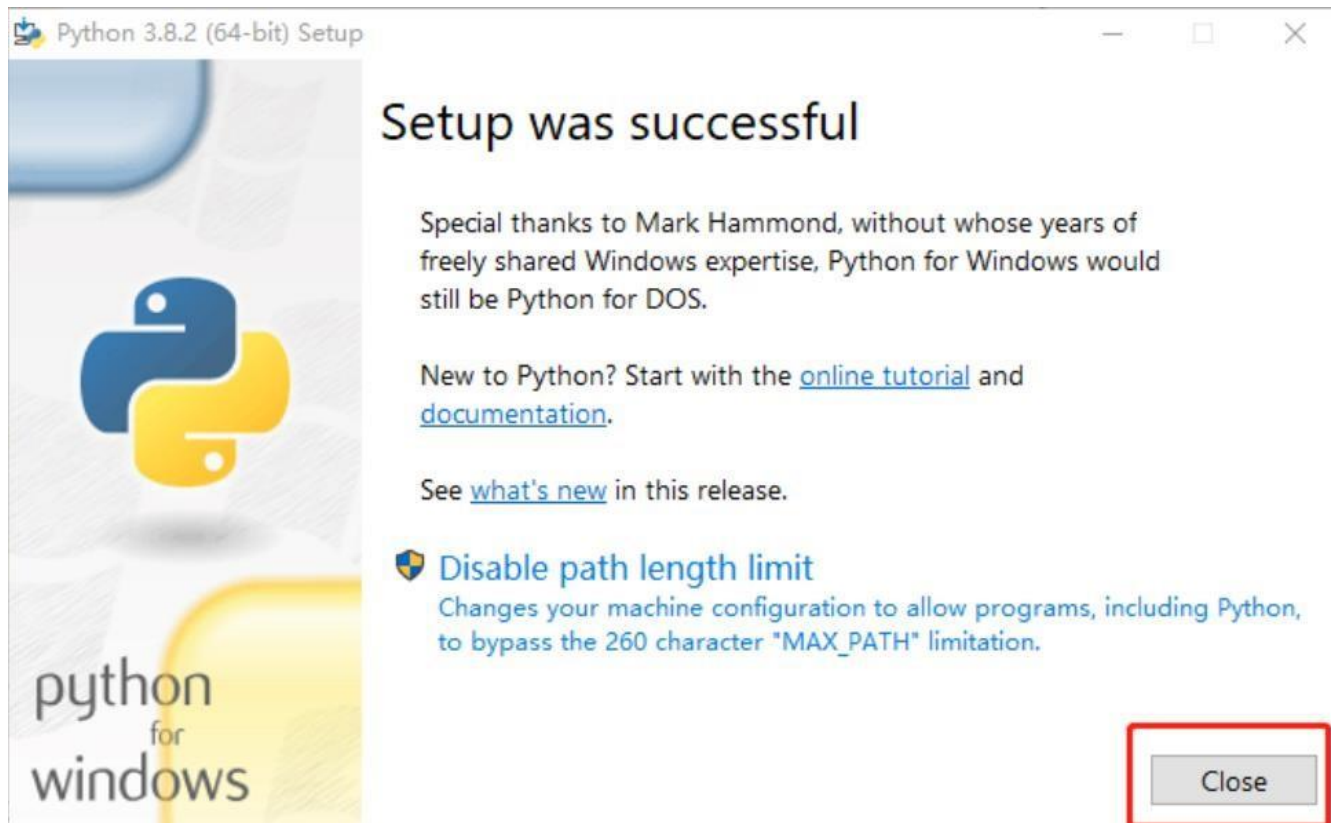
# 1.4 Python简介

## 1.4.4 Python安装和使用



# 1.4 Python简介

## 1.4.4 Python安装和使用



# 1.4 Python简介

## 1.4.4 Python安装和使用

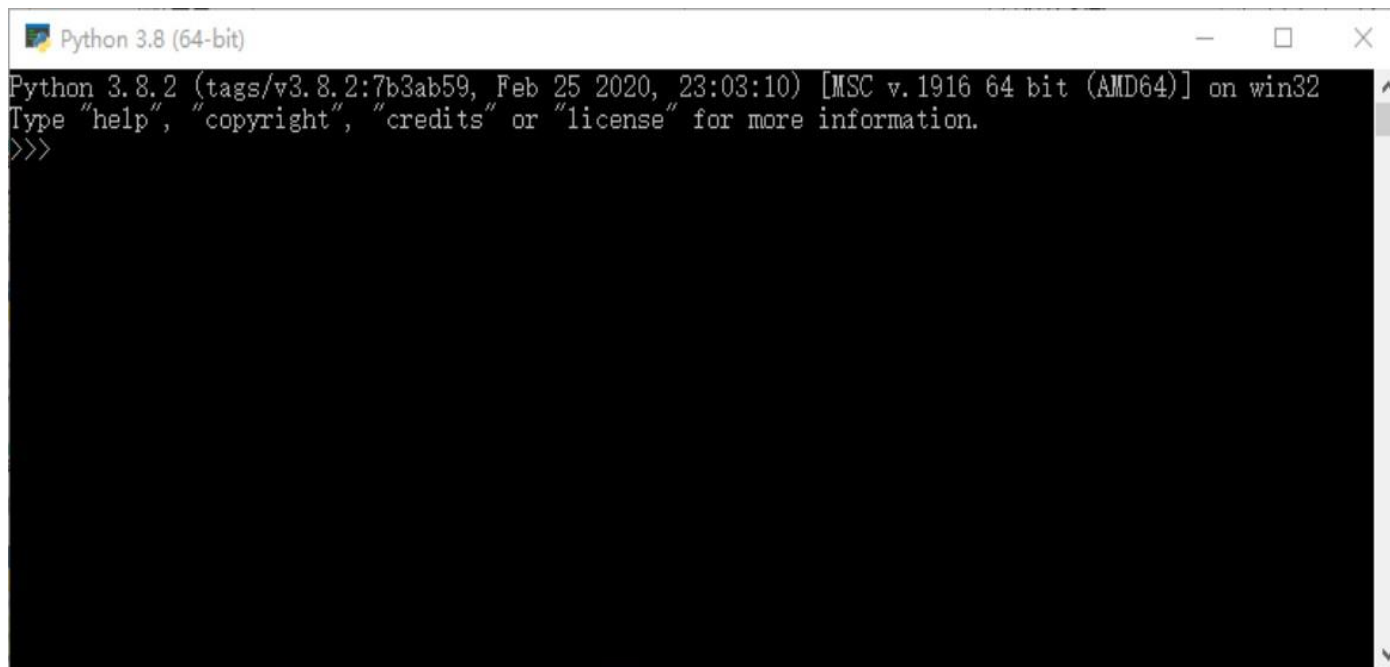
- 点击Windows左下角【开始】菜单
- 找到【Python3.8】



# 1.4 Python简介

## 1.4.4 Python安装和使用

- 进入命令行界面



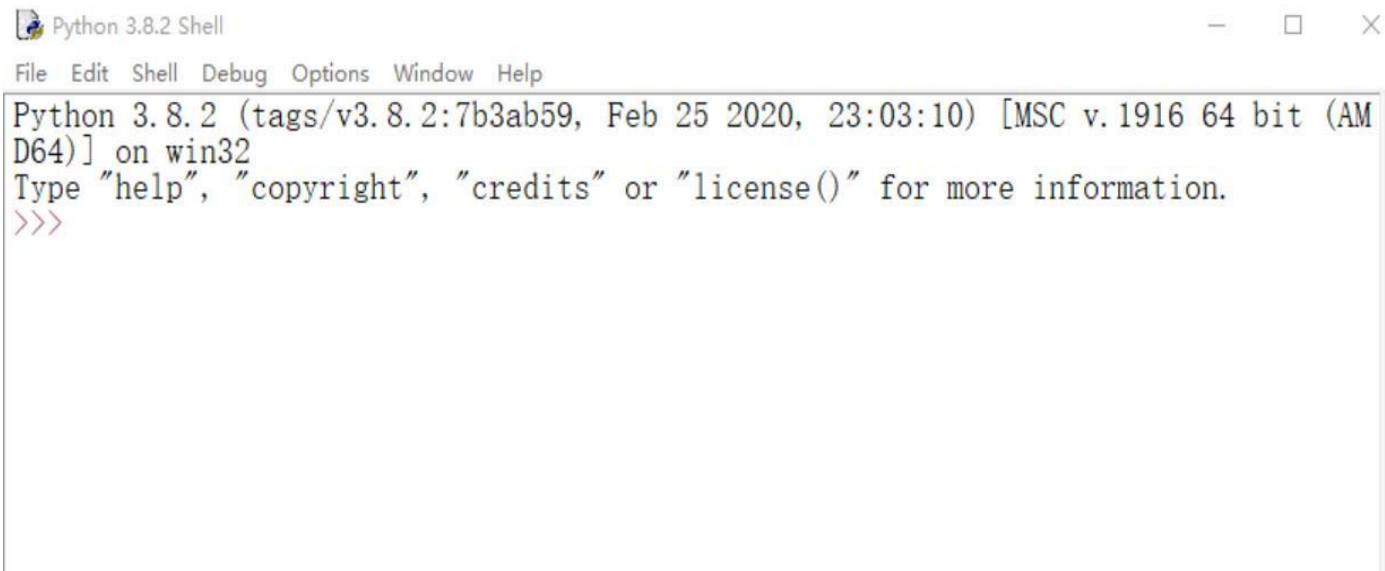
```
Python 3.8 (64-bit)
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```



# 1.4 Python简介

## 1.4.4 Python安装和使用

- 进入IDLE界面

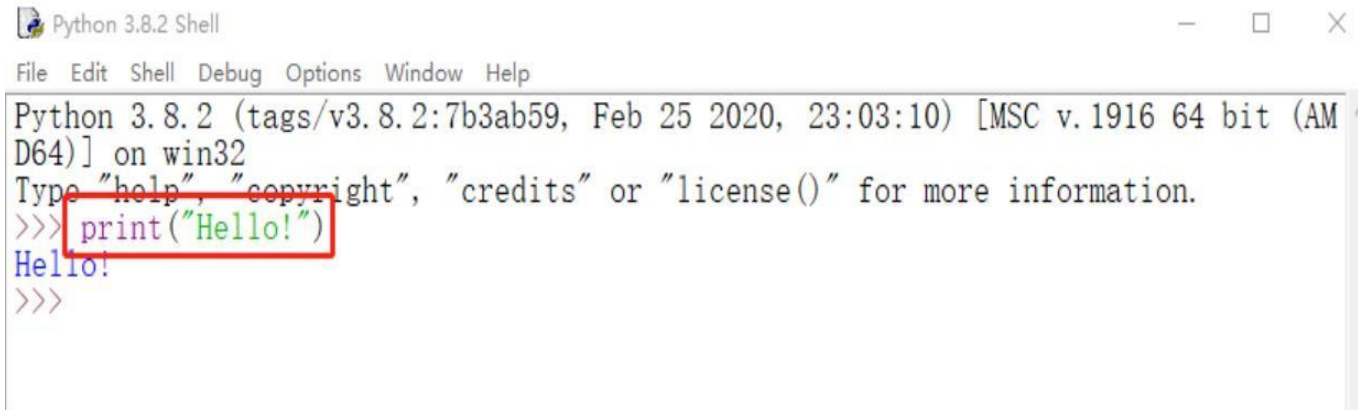


```
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
```

# 1.4 Python简介

## 1.4.4 Python安装和使用

- 在Python shell界面下打印第一行语句



The screenshot shows a window titled "Python 3.8.2 Shell". The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main text area displays the following content:

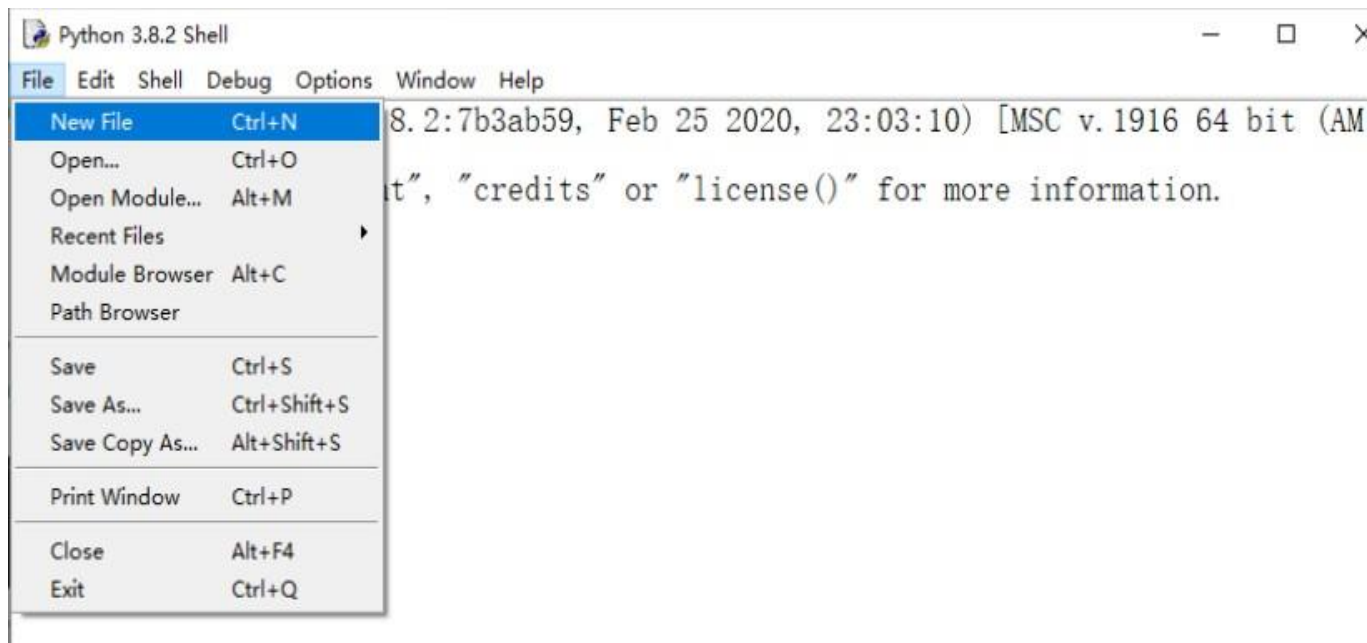
```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("Hello!")
Hello!
>>>
```

The line `print("Hello!")` is highlighted with a red rectangular box.

# 1.4 Python简介

## 1.4.4 Python安装和使用

- 在Python shell界面下新建一个程序文件：【File】→【New File】



# 1.4 Python简介

## 1.4.4 Python安装和使用

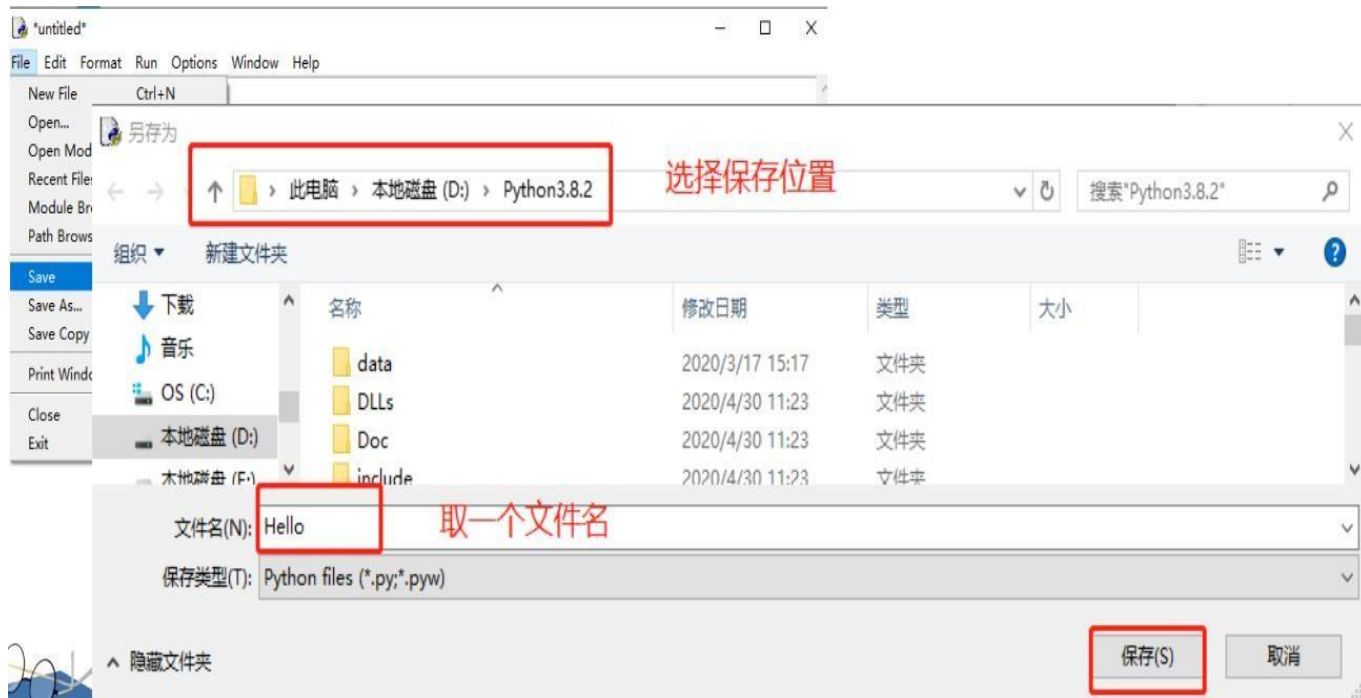
- 在新建的程序文件内输入代码



# 1.4 Python简介

## 1.4.4 Python安装和使用

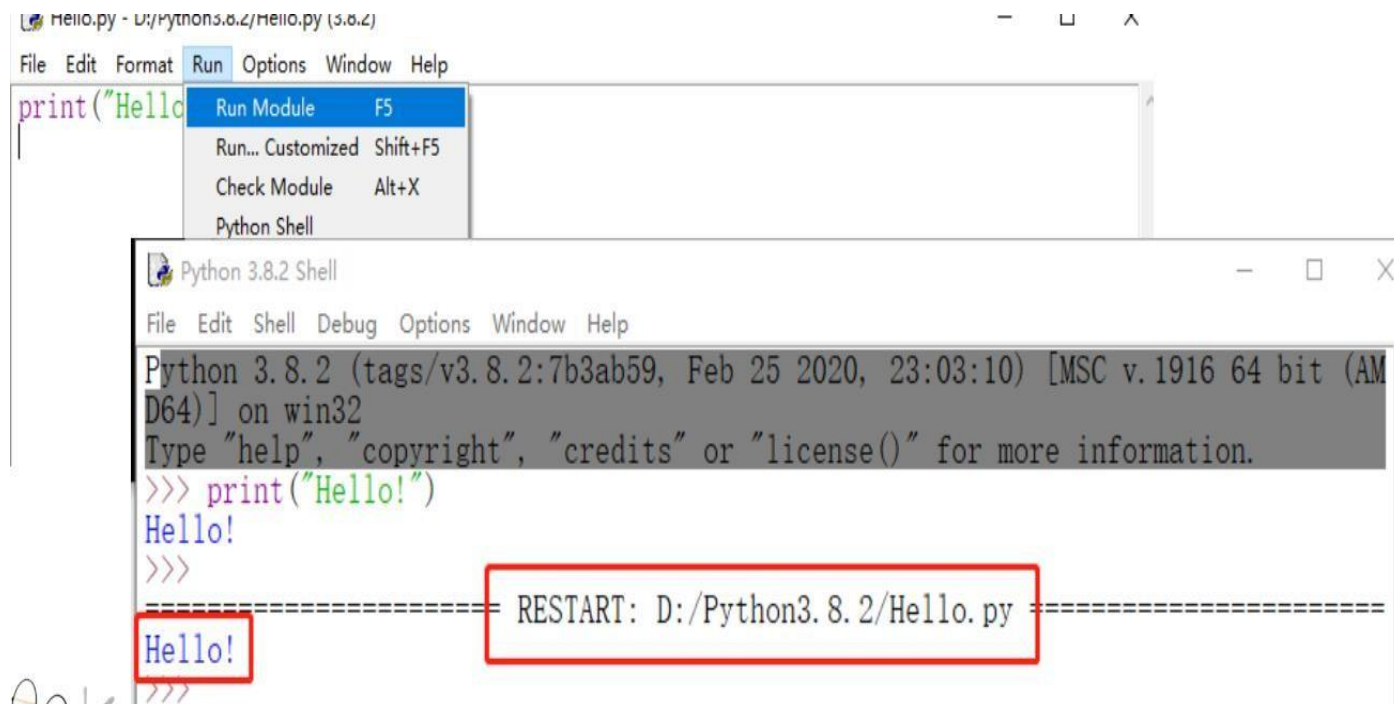
- 保存文件



# 1.4 Python简介

## 1.4.4 Python安装和使用

- 运行程序和查看结果



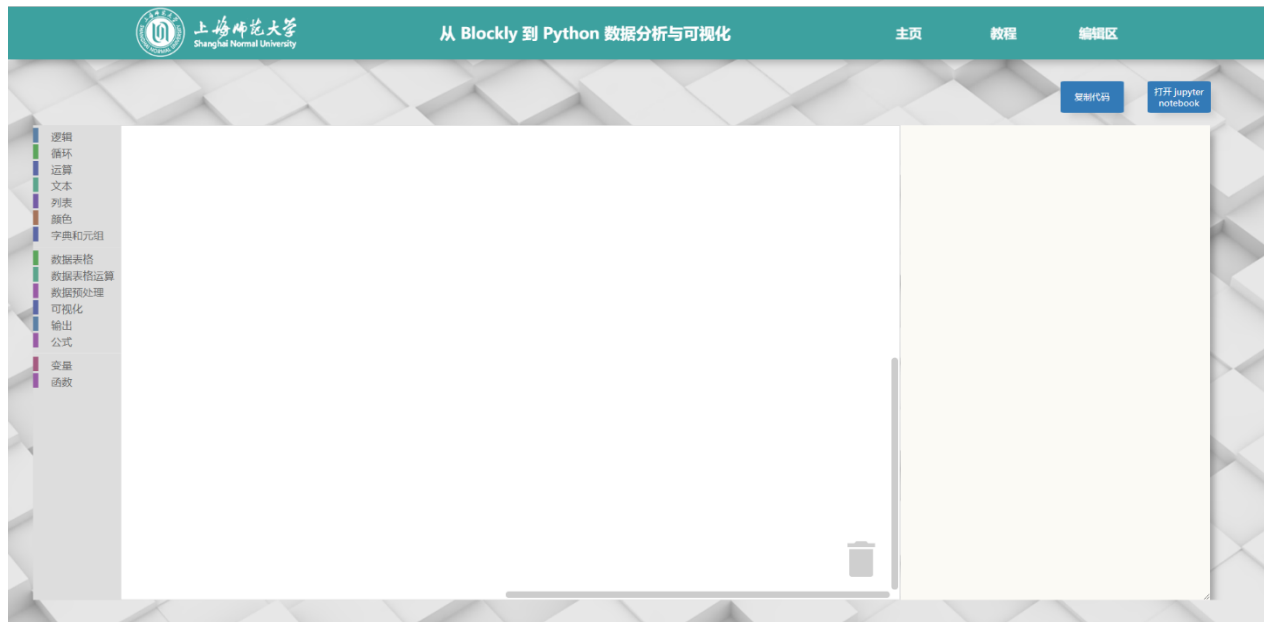
The screenshot shows a Python IDE window titled "Hello.py - D:/Python3.8.2/Hello.py (3.8.2)". The menu bar includes File, Edit, Format, Run, Options, Window, and Help. The "Run" menu is open, showing options: Run Module (F5), Run... Customized (Shift+F5), Check Module (Alt+X), and Python Shell. Below the IDE window, a "Python 3.8.2 Shell" window is open, displaying the following text:

```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("Hello!")
Hello!
>>>
```

Below the shell window, a red box highlights the text "Hello!". To the right, another red box highlights the text "RESTART: D:/Python3.8.2/Hello.py".

# 1.5 网站介绍

- <http://blockly.chajiuqq.cn/>
- 本网站基于 Blockly 可视化拖拽的编程方法，设计了一套从 Blockly 到 Python 数据分析与可视化教学平台及案例。
- 本网站教学内容非常适合中学阶段的物理、化学、数学、生物等实验数据处理与可视化。
- 通过本网站的学习，学生可以掌握基本的数据分析与可视化的概念与方法。



## 1.6 blockly生成第一个“Hello world”程序

- 逻辑
- 循环
- 运算
- 文本
- 列表
- 颜色
- 字典和元组
- 数据表格
- 数据表格运算
- 数据预处理
- 可视化
- 输出
- 公式
- 变量
- 函数

A Blockly 'say' block, which is a blue block with a 'say' label and a text input field containing 'hello world!'.

```
print('hello world! ')
```



# 1.7 本章小结:

本章主要学习了Blockly的概念、模块功能，学会如何搭建Blockly编程环境以及初步掌握Python的安装和使用。如何使用 <http://blockly.chajiugqq.cn/> 进行可视化编程 是本章的重点。

