

# Selenium 自动化测试实战



扫码试看/订阅

《Selenium自动化测试实战》视频课程

# 聊聊 Web 自动化测试

1. 什么是测试
2. 测试的分类
3. 什么是自动化测试
4. 自动化测试的技术选型

# 什么是测试

- 首先，我们来看什么是测试：
  - 通俗的来讲，程序测试就是 运行程序，并发现程序的错误。
  - 专业一点来讲就是：验证软件的正确性、完整性、安全性和质量的过程。
  - 用程序员的话说就是找 bug。
- 这就是软件测试，软件测试是一个过程。

# 测试的分类



# 什么是自动化测试

- 自动化测试就是，假定一些预设值，由机器自动化完成的测试称为自动化测试，如下图所示。



- Web 自动化测试，将自动化测试技术应用到 Web 测试中，是假定一些预设值，由程序驱动 Web 浏览器来完成 Web 程序的测试。如下图所示：



# Web 自动化测试的技术选型

- 目前，Web 自动化测试工具，可以说种类繁多，五花八门。选择标准应该从以下几方面来考虑：
  - 发展趋势要稳定上升
  - 要有一个活跃的社区能够讨论
  - 要是开源软件
  - 支持主流的编程语言（Java、Python）
  - 要支持主流浏览器（谷歌、IE、火狐等）
- 综合以上几个方面考虑，本套课程采用的技术架构是：Python+Selenium。

# Selenium 三剑客



# Selenium 三剑客

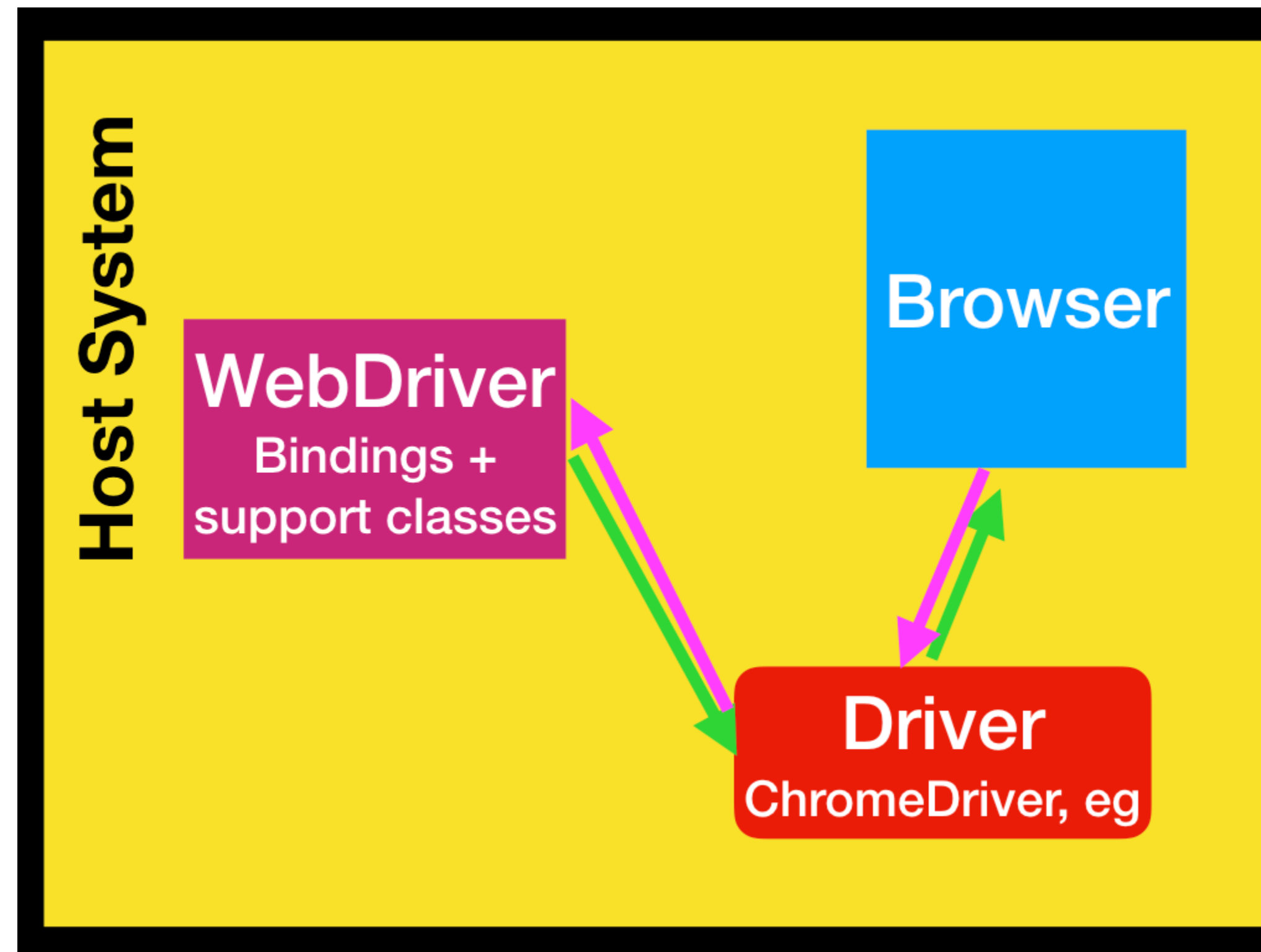
- Selenium 是一个用于 Web 应用程序自动化测试工具。Selenium 测试直接运行在浏览器中，就像真正的用户在操作一样。
- 主要功能包括：
  - 测试与浏览器的兼容性——测试你的应用程序看是否能够很好的工作在不同浏览器和操作系统之上。
  - 测试系统功能——创建回归测试检验软件功能和用户需求。

# Selenium 三剑客

- 我们通常所说的 Selenium 是对 Selenium 的总称，其中包括三个项目，分别是：
  - Selenium WebDriver
  - Selenium IDE
  - Selenium Grid
- 下面分别来介绍这三个项目。

# Selenium WebDriver

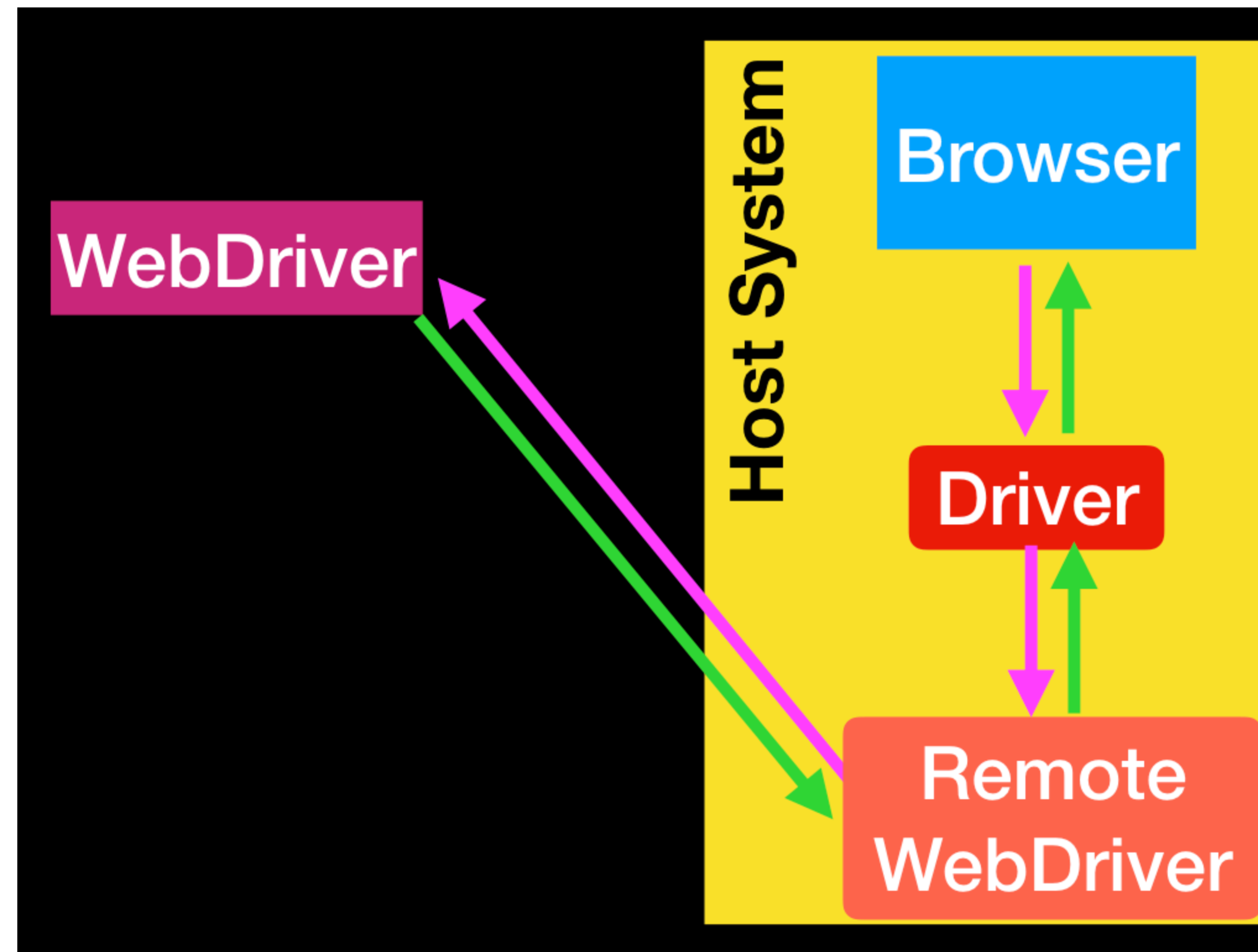
- Selenium WebDriver 是客户端 API 接口，测试人员通过调用这些接口，来访问浏览器驱动，浏览器驱动再访问浏览器。
- 过程如下图所示：



# Selenium WebDriver

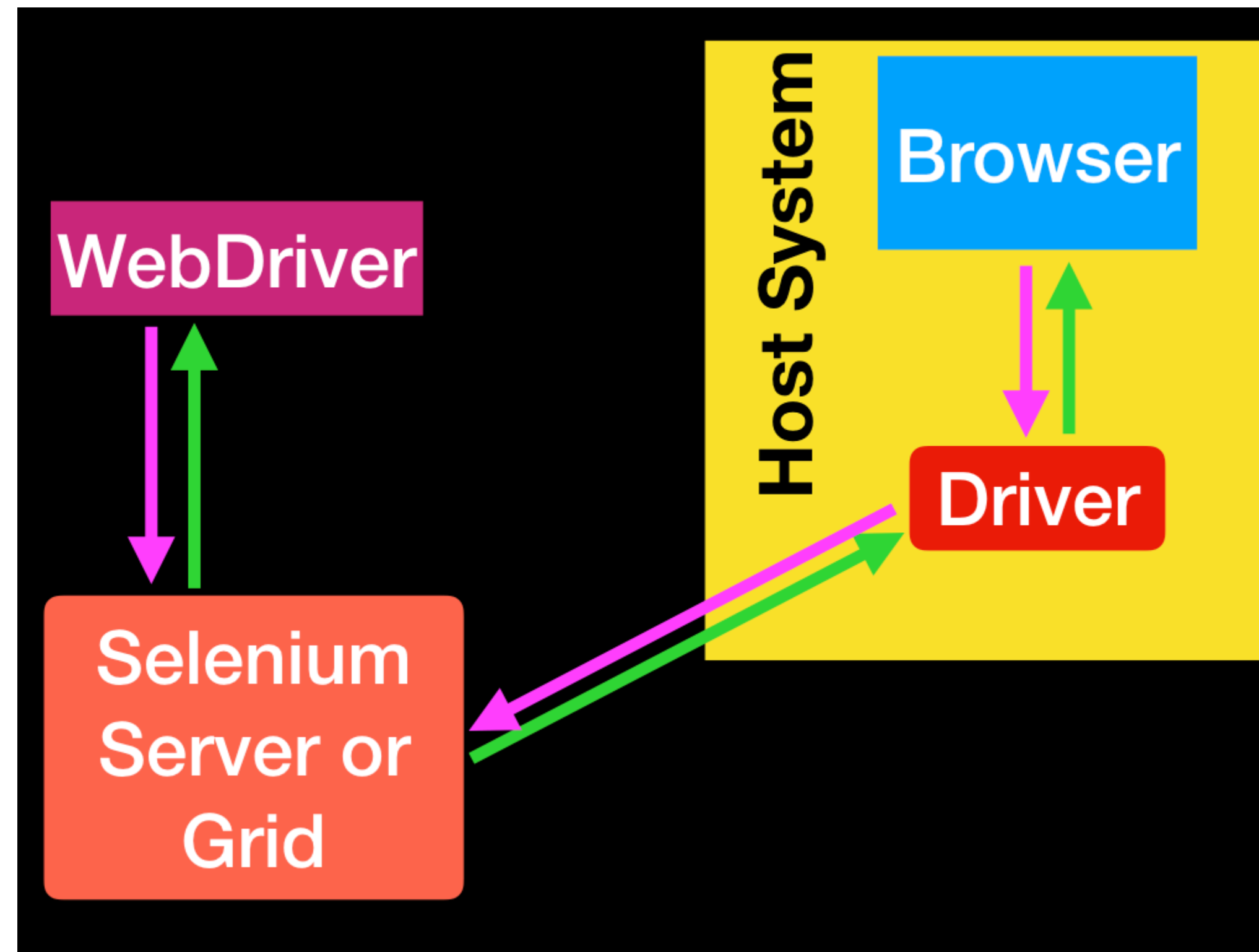
- 另外，与浏览\*器的\*通信也可以是通过 Selenium Server 或 RemoteWebDriver 的\*远程\*通信。RemoteWebDriver 与驱动程序和浏览器在同一系统上运行。

- 如下图所示：



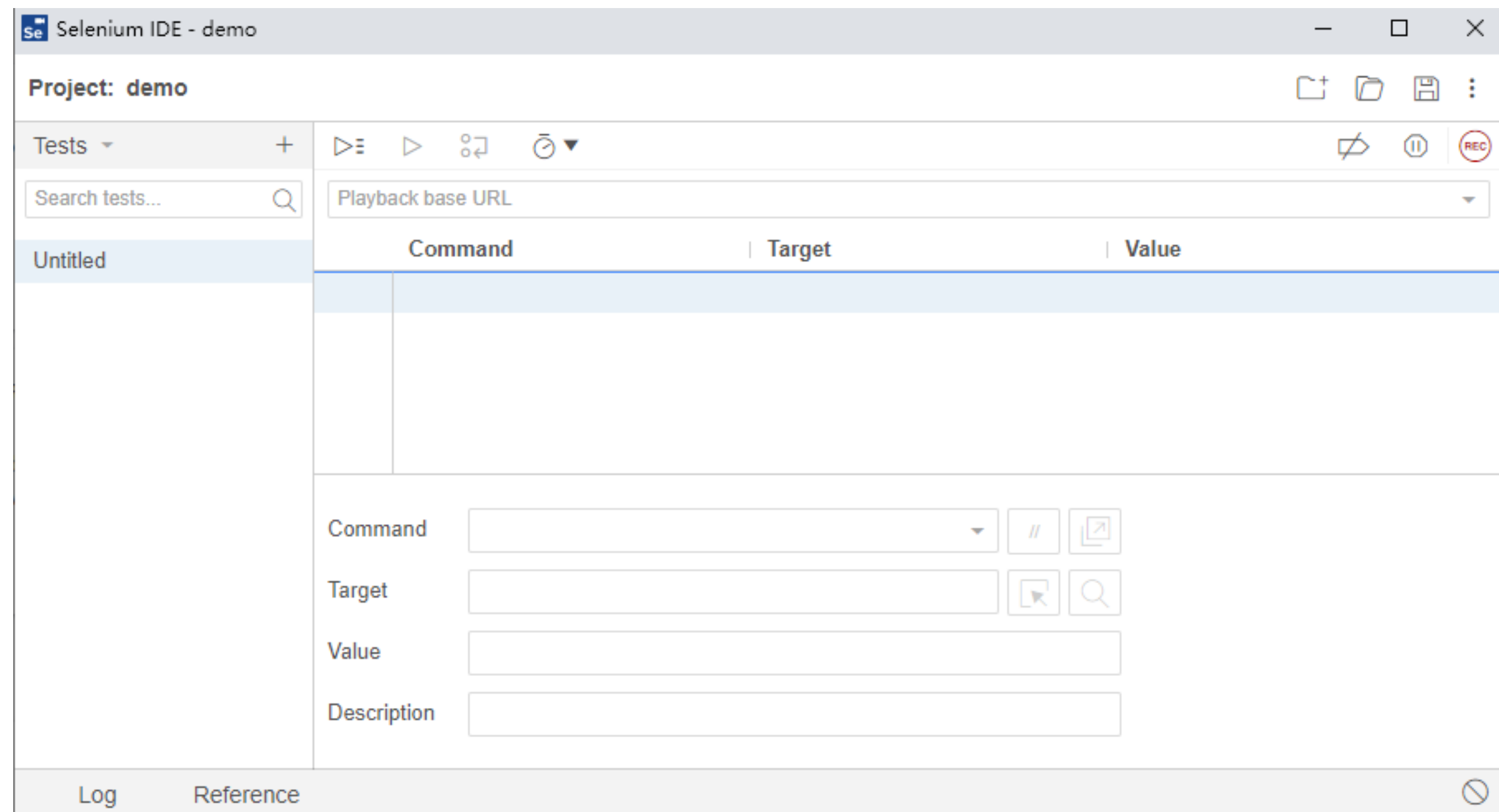
# Selenium WebDriver

- 除此之外，还可以使用 Selenium Server 或 Selenium Grid 进行分布式测试。
- 如下图所示：



# Selenium IDE

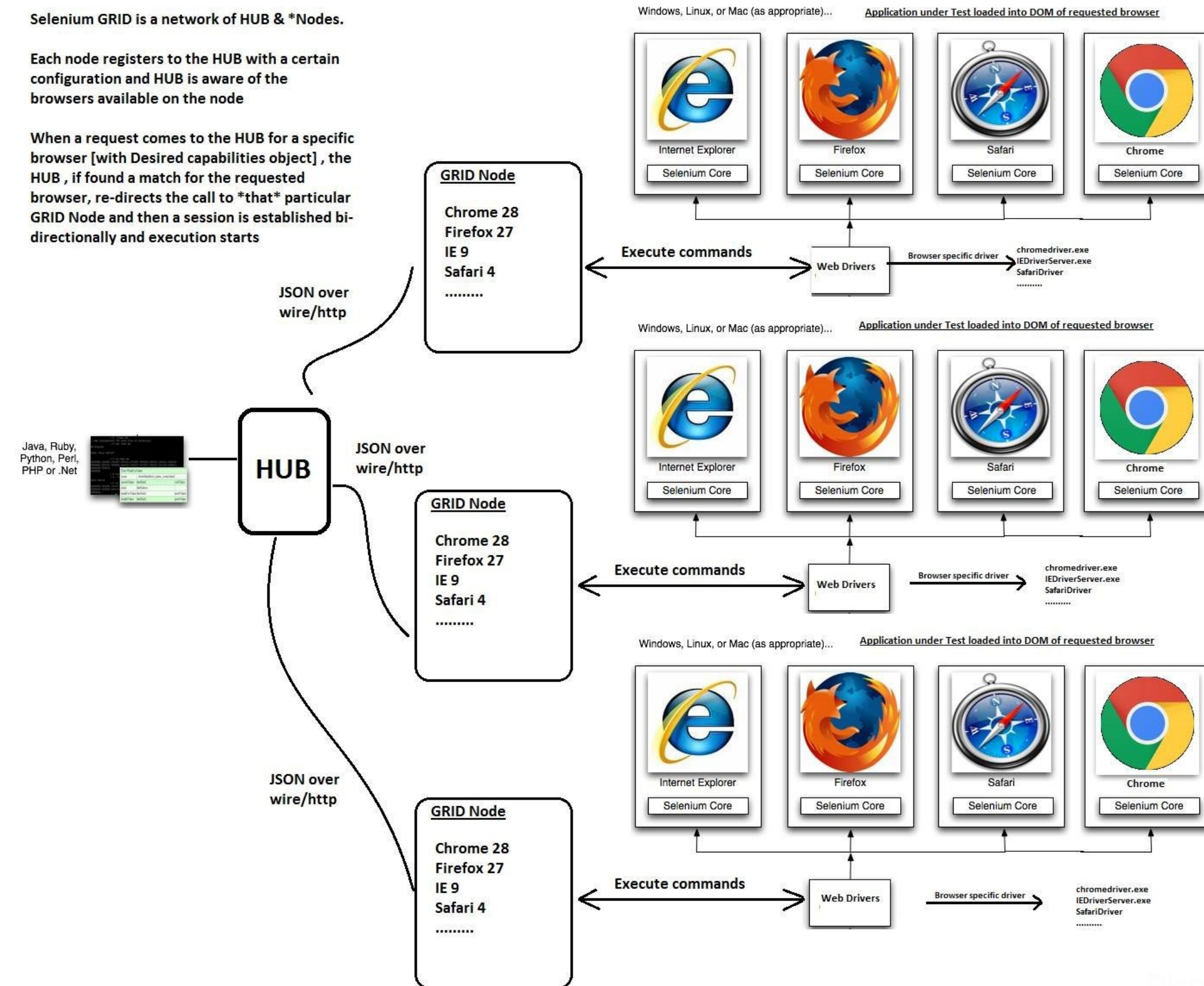
- Selenium IDE 是一个浏览器插件，支持 Chrome 浏览器和 Firefox 浏览器，可以将手动测试过程记录下来，并生产自动化测试脚本，可以实现回放，对于初学者，学习脚本和回归测试都很有帮助。运行结果如下。我们也可以打开浏览器来体验一下。有个 Selenium IDE 如何使用我们会在后面章节详解讲解。



# Selenium Grid

- 如果你想通过多台计算机上进行分布式来扩容，并从一个中心点管理多个环境，从而轻松地对多种浏览器/OS 组合运行测试，那么可以使用 Selenium Grid。

- Selenium Grid的结构如下图所示：





# Selenium的特点

- 开源，免费
- 多浏览器支持：Firefox、Chrome、IE、Opera、Edge
- 多平台支持：Linux、Windows、MAC
- 多语言支持：Java、Python、Ruby、C#、JavaScript、C++
- 支持分布式测试（使用 Selenium Grid）
- 支持录制、回放和脚本生成（使用 Selenium IDE）



下载安装 Python 和 Selenium

# 下载安装 Python和 Selenium

- 本课程我们使用 Python 作为客户端语言，来操作 WebDriver。大部分测试环境都是 Windows，这里就以 Windows 为例。
- Windows 下安装 Python 需要如下几个步骤：
  - 下载 Python
  - 安装 Python
  - 设置 Python的环境变量
  - 安装 Selenium

# 下载 Python

- 进入 Python 的官网：<https://python.org/>
- 选择 “Downloads” 菜单项，Python 的版本分为2和3，这里使用 Python3.7.6。
- 下载地址为：<https://www.python.org/ftp/python/3.7.6/python-3.7.6-amd64.exe>

# 安装 Python

- Python 的安装非常简单，一路下一步就可以安装成功，下面我们来演示一下。

# 设置 Python 环境变量

- 选择 Python 的安装路径，定位到 python.exe 所在的目录，例如：C:\python3.7，将该目录添加到系统环境变量，选择我的电脑->属性->高级系统设置->环境变量->选择系统环境变量->Path->添加->将C:\python3.7添加到环境变量。
- 按照同样的方法设置pip.exe环境变量， pip.exe所在的文件夹为：  
C:\python3.7\Scripts\pip.exe

# 设置 Python 环境变量

- 经过以上两个步骤，下面我们来测试一下，打开命令行 cmd，输入 python，如果出现如下窗口表示安装成功：

```
命令提示符 - python
Microsoft Windows [版本 10.0.18362.900]
(c) 2019 Microsoft Corporation。保留所有权利。

C:\Users\ghz>python
Python 3.7.6 (tags/v3.7.6:43364a7ae0, Dec 19 2019, 00:42:30) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information
>>>
```

```
命令提示符
Microsoft Windows [版本 10.0.18362.900]
(c) 2019 Microsoft Corporation。保留所有权利。

C:\Users\ghz>pip
Usage:
  pip <command> [options]

Commands:
  install           Install packages.
  download          Download packages.
  uninstall         Uninstall packages.
  freeze            Output installed packages in requirements file.
  list              List installed packages.
  show              Show information about installed packages.
  check             Verify installed packages have compatible dependencies.
  help              Display this help message.
```

# 安装 Selenium

- pip 是 Python 的模块管理工具，可以方便的安装卸载模块，Selenium 就是 Python 的一个模块，所以，可以方面的使用 pip 来安装 Selenium:
- `pip install selenium`
- 在命令行输入，如下命令查看 Selenium 的安装情况:
- `pip list`
- 如下图所示:



```
选择命令提示符
'selenium' 不是内部或外部命令，也不是可运行的程序
或批处理文件。

C:\Users\ghz>pip list
Package      Version
-----
autopep8     1.4.4
mccabe       0.6.1
pip          19.2.3
pycodestyle  2.5.0
selenium     3.141.0
setuptools   41.2.0
urllib3      1.25.9
WARNING: You are using pip version 19.2.3, however version 20.1.1 is
available.
You should consider upgrading via the 'python -m pip install --upgrad
e pip' command.

C:\Users\ghz>
```

下载安装 Pycharm



# 下载安装 PyCharm

- PyCharm 是 JetBrains 公司开发的，专门用来开发 Python 应用的 IDE，本课程使用 PyCharm 作为开发工具。
- 使用 PyCharm 需要如下几个步骤：
  - 下载 PyCharm
  - 安装 PyCharm
  - 使用 PyCharm

# 下载 PyCharm

- PyCharm 的下载地址为：  
<https://www.jetbrains.com/pycharm/download/#section=windows>
- PyCharm 的版本分为专业版和社区版，专业版功能比较强大但是是付费的，社区版是免费的，本课程使用社区版可以完全胜任测试程序的开发。

# 安装 PyCharm

- PyCharm 的安装非常简单，没有什么需要特殊设置的地方，一路“下一步”就可以安装成功，我们来演示一下。

# 使用 PyCharm

- 使用 PyCharm 创建项目。
- 使用 PyCharm 创建 Python 模块
- 运行 Python 模块
- 使用 Python Console
- 下面我们来演示一下。

# 安装 Selenium

- pip 是 Python 的模块管理工具，可以方便的安装卸载模块，Selenium 就是 Python 的一个模块，所以，可以方面的使用 pip 来安装 Selenium:
- `pip install selenium`
- 在命令行输入，如下命令查看 Selenium 的安装情况:
- `pip list`
- 如下图所示:



```
选择命令提示符
'selenium' 不是内部或外部命令，也不是可运行的程序
或批处理文件。

C:\Users\ghz>pip list
Package      Version
-----
autopep8     1.4.4
mccabe       0.6.1
pip          19.2.3
pycodestyle  2.5.0
selenium     3.141.0
setuptools   41.2.0
urllib3      1.25.9
WARNING: You are using pip version 19.2.3, however version 20.1.1 is
available.
You should consider upgrading via the 'python -m pip install --upgrad
e pip' command.

C:\Users\ghz>
```

下载安装浏览器驱动

# 下载安装浏览器驱动

- Selenium 需要安装对应的浏览器驱动，才能驱动浏览器自动测试。下载安装驱动需要如下几个步骤：
  1. 检查浏览器版本
  2. 下载对应版本驱动
  3. 设置驱动环境变量
  4. 测试驱动

# 检查浏览器版本

- 驱动版本必须和浏览器的版本对应才可以。
- 查看 Chrome浏览器版本，可以在 URL 地址栏输入：  
[chrome://version/](chrome://version/) 进行查看，也可以点击菜单栏->帮助  
->关于 Google Chrome 查看。
- 查看 Firefox 浏览器版本，选择菜单栏 Help->About Firefox。



# 下载对应版本驱动

- 打开Selenium官网：<https://www.selenium.dev/>
- 选择文档 Documentation
- 选择 Selenium 安装 [https://www.selenium.dev/documentation/zh-cn/selenium\\_installation/](https://www.selenium.dev/documentation/zh-cn/selenium_installation/)
- 选择安装 WebDriver 二进制文件 [https://www.selenium.dev/documentation/zh-cn/selenium\\_installation/installing\\_webdriver\\_binaries/](https://www.selenium.dev/documentation/zh-cn/selenium_installation/installing_webdriver_binaries/)
- 选择 WebDriver 二进制文件 [https://www.selenium.dev/documentation/zh-cn/webdriver/driver\\_requirements/](https://www.selenium.dev/documentation/zh-cn/webdriver/driver_requirements/)
- 6. 找到快速参考，如下图所示：

浏览器	支持的操作系统	维护者	下载	问题追踪
Chromium/Chrome	Windows/macOS/Linux	谷歌	<a href="#">下载</a>	<a href="#">问题</a>
火狐	Windows/macOS/Linux	Mozilla	<a href="#">下载</a>	<a href="#">问题</a>
Edge	Windows 10	微软	<a href="#">下载</a>	<a href="#">问题</a>
Internet Explorer	Windows	Selenium 项目组	<a href="#">下载</a>	<a href="#">问题</a>
Safari	macOS El Capitan 及更高版本	苹果	内置	<a href="#">问题</a>
Opera	Windows/macOS/Linux	Opera	<a href="#">下载</a>	<a href="#">问题</a>

# 设置环境变量

- 驱动路径可以在编写程序时设置，也可以设置为系统环境变量，程序会自动找到。为了方便，我们设置为环境变量。
- 将下载的驱动统一放到一个文件夹中
- 将该文件添加到系统环境变量

# 测试驱动

- 打开命令行，输入驱动名称，例如：chromedriver.exe（Chrome驱动）
- 如下图所示：

```
命令提示符 - chromedriver
Microsoft Windows [版本 10.0.18362.900]
(c) 2019 Microsoft Corporation. 保留所有权利。

C:\Users\ghz>chromedriver
Starting ChromeDriver 83.0.4103.39 (ccbf011cb2d2b19b506d844400483861342c20cd-refs/branch-heads/4103@{#416}) on port 9515
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations
for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
```

下载安装JDK

# 下载安装JDK

- 我们的 Selenium Grid 分布式测试，使用的服务器是使用 Java 编写的一个 jar 包程序，运行该程序需要 Java 运行环境，这里我们需要安装 JDK。
- JDK 安装步骤如下：
  1. JDK 下载
  2. JDK 安装
  3. 环境变量配置
  4. 测试 JDK 安装

# JDK 下载

- 下载地址：
  - <https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html>
- 下载需要注册一个 Oracle 账号，登录后可以下载。下面我们来演示一下。

# 安装 JDK

- JDK 安装非常简单，基本上是一路点击“下一步”即可完成，下面我们来演示一下。

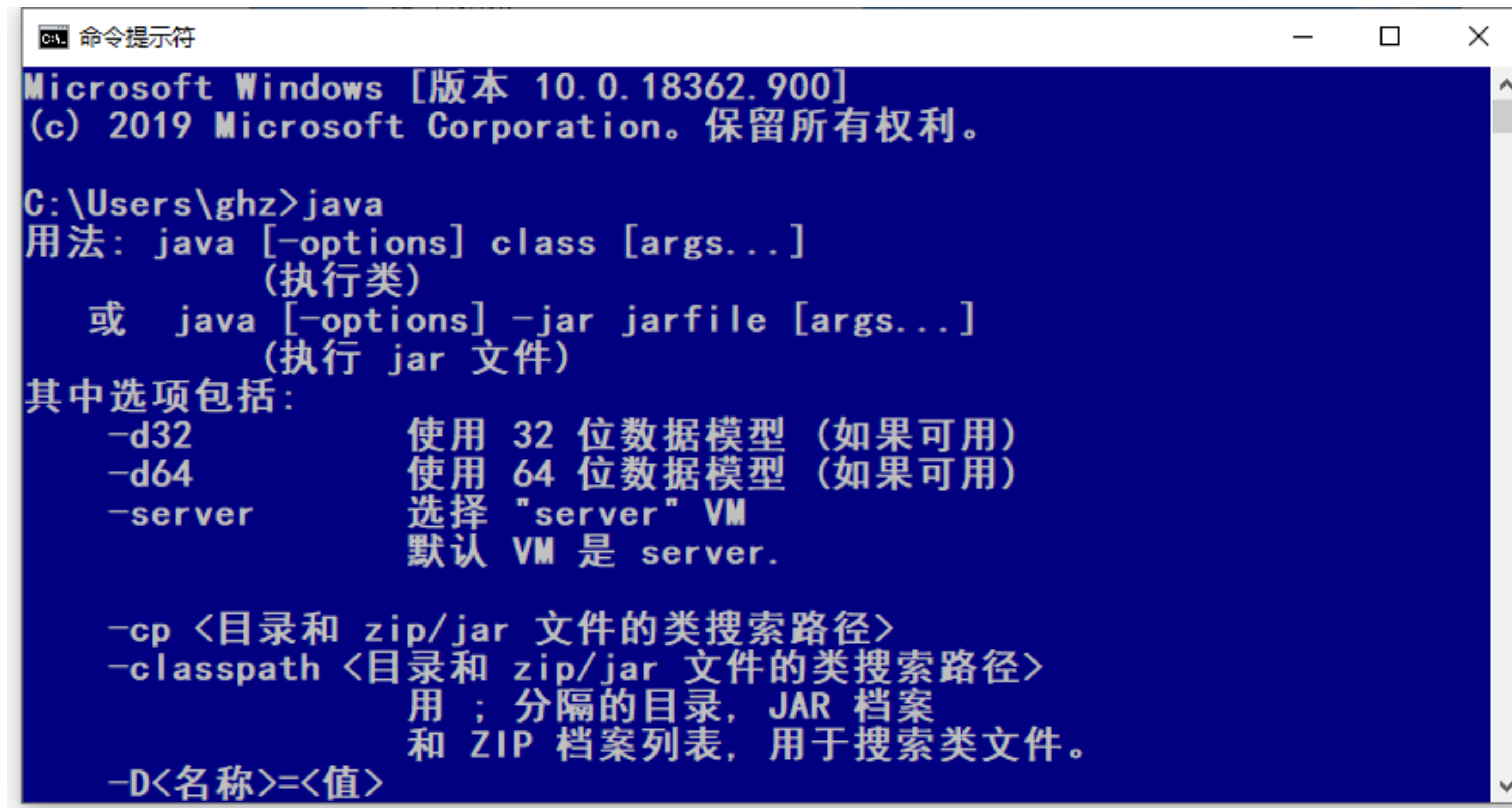
# 环境变量配置

- 为了在任何地方都能使用javac.exe 和java.exe这些命令，我们需要配置环境变量，下面我们来演示一下。



# 测试 JDK 安装

- 打开命令行输入 Java，如果出现如下图所示内容表示安装成功。



```
命令提示符
Microsoft Windows [版本 10.0.18362.900]
(c) 2019 Microsoft Corporation。保留所有权利。

C:\Users\ghz>java
用法: java [-options] class [args...]
           (执行类)
 或 java [-options] -jar jarfile [args...]
           (执行 jar 文件)
其中选项包括:
  -d32          使用 32 位数据模型 (如果可用)
  -d64          使用 64 位数据模型 (如果可用)
  -server       选择 "server" VM
                默认 VM 是 server.

  -cp <目录和 zip/jar 文件的类搜索路径>
  -classpath <目录和 zip/jar 文件的类搜索路径>
                用 ; 分隔的目录, JAR 档案
                和 ZIP 档案列表, 用于搜索类文件。
  -D<名称>=<值>
```

第一个 Selenium 测试

# 第一个 Selenium 测试

- 这一节我们来创建一个简单的 Selenium 应用，这个应用非常简单，就是通过 Selenium API 来实现自动打开浏览器。程序代码如下：

```
from selenium import webdriver
```

```
from time import sleep
```

```
driver = webdriver.Chrome()
```

```
driver.get('http://www.baidu.com')
```

```
sleep(2)
```

```
driver.quit()
```

下面通过分析源码的方式来理解一下这段代码的执行过程。

# WebDriver 分析

- 程序的运行结果也非常简单，就是打开 Chrome 浏览器，定位到百度首页，即：  
`http://www.baidu.com`，2秒钟后，浏览器退出。
- 下面我们来分析一下这一段程序，第一行 `from selenium import webdriver`，这一行是导入包，从 Selenium 这个包中导入 WebDriver 这个包。
- 从 Pycharm 的扩展库（External Libraries）下面找到 Selenium，下面有两个子包 Common和 WebDirver 和一个包初始化文件 `__init__.py`。
- 打开 WebDriver 这个包，查看一下 `__init__.py`，这个包初始化文件就是导入子包中的一些类，并进行重命名。

# driver = webdriver.Chrome()源码分析

- `driver = webdriver.Chrome()` 这行代码是实例化一个 Chrome WebDriver, 我们来分析一下他的初始化方法, 查看源码。
- `executable_path` 是驱动的路径, 可以在环境变量中设置, 也可以实例化时指定。默认值是: ChromeDriver 就是我们设置在环境变量中的 `chromedriver.exe` 注意: 这个名字要对应。如果我们把系统目录中的驱动名称修改后, 程序将会找不到驱动, 而报错。
- `port` 是端口号, 0 表示随机端口号, 后面的几个参数我们在后面的课程中再来介绍。

# Service 源码分析

- 接下来我们分析下面的代码：

```
self.service = Service(  
  
    executable_path,  
  
    port=port,  
  
    service_args=service_args,  
  
    log_path=service_log_path)  
  
self.service.start()
```

- 这个代码是实例化一个 Service，并且调用 start 方法。我们再来分析一下start方法。

# Service 源码分析

- 这个方法就是通过 subprocess 模块的 Popen 方法调用 chromedriver.exe。类似下面的代码：

```
import subprocess
```

```
def f():
```

```
    p = subprocess.Popen("chromedriver")
```

```
    p.communicate()
```

```
if __name__ == '__main__':
```

```
    f()
```



扫码试看/订阅

《Selenium自动化测试实战》视频课程