

.NET Conf China
2022



我眼中的 .NET

袁诗琪



.NET Conf China



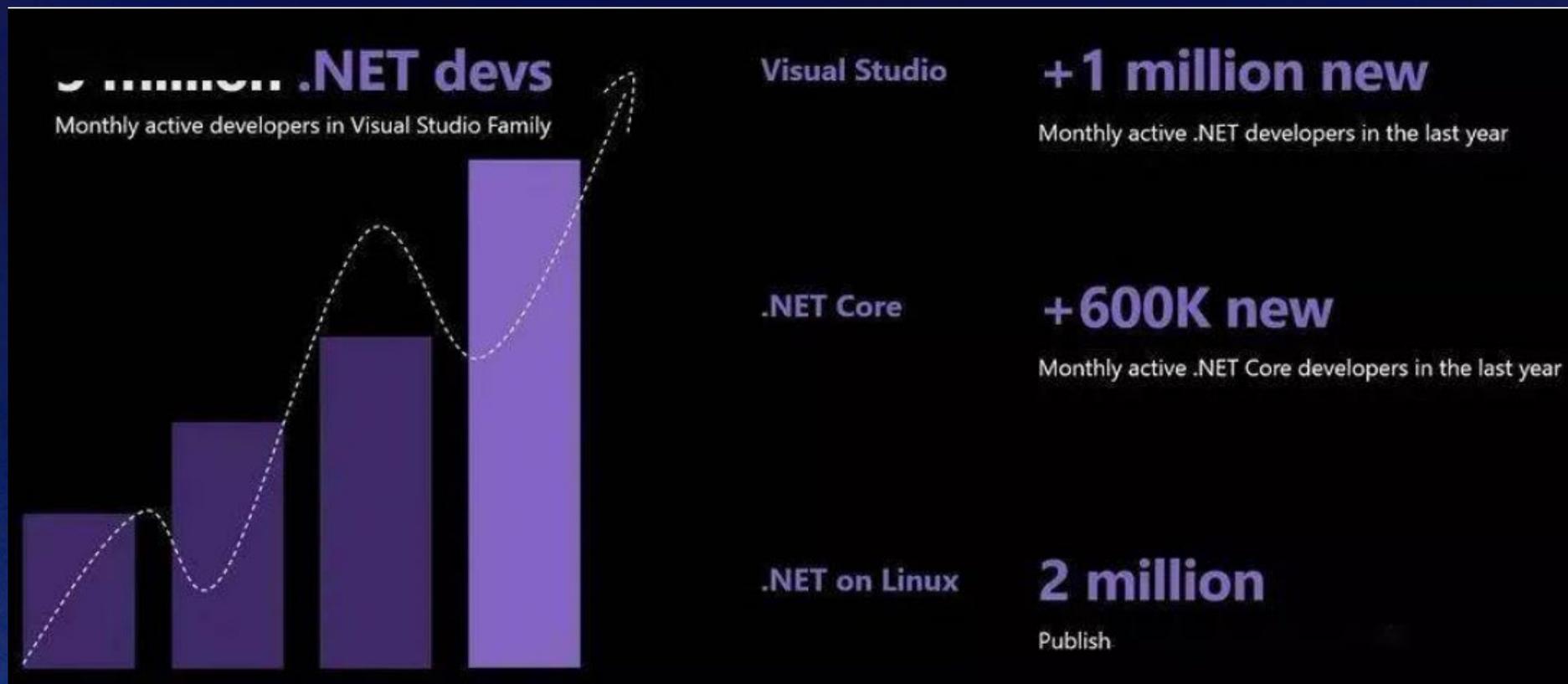
.NET 的现状和未来



.NET的现状



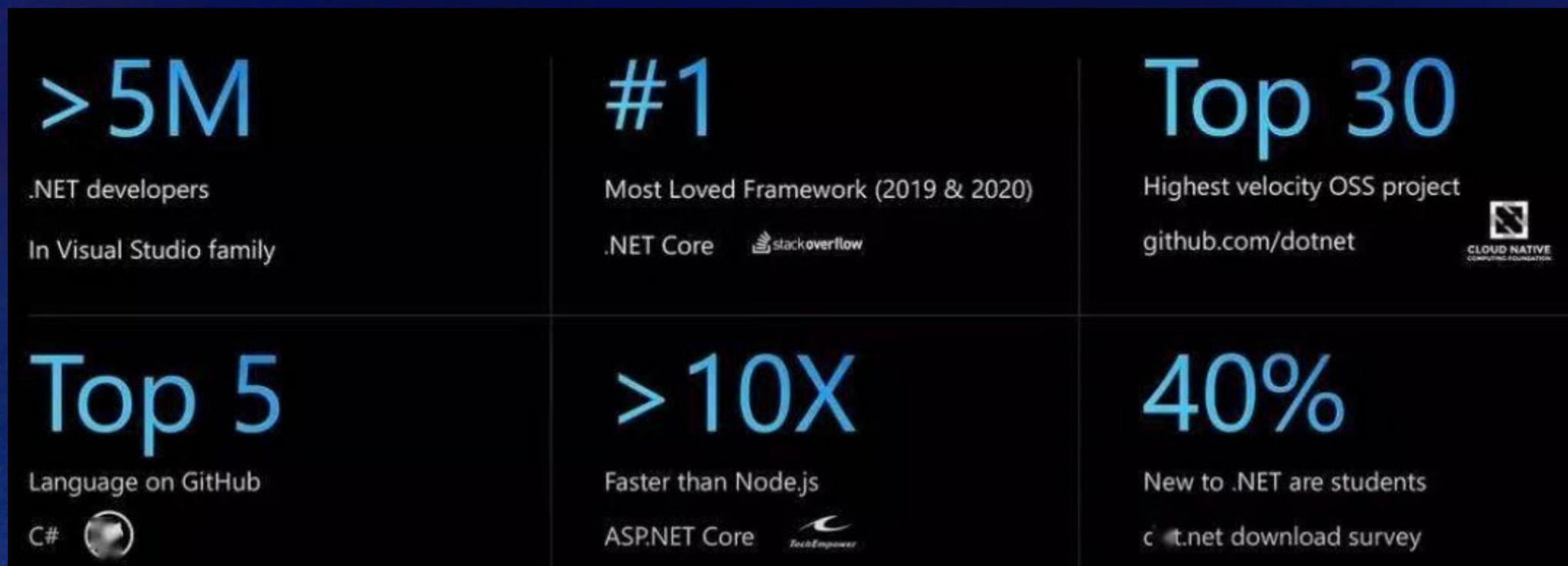
关于现状，我从官方的几个 YouTube 视频介绍中截取了两张图。一张是 .NET 平台截止 2020 年 12 月开发者数量大致的变化情况：



.NET的现状



另一张是 .NET 平台截止 2021年 5 月开发者数量、受欢迎度、性能等方面的表现数据：



.NET的现状



开发者数量

截止 2021年 5 月，全球有超过 500 万的 .NET 开发者。过去两年的跨平台 .NET Core 活跃开发者数量增长超过 60 万。而这些数据仅来源于 Visual Studio IDE，不包含使用其它 IDE 的开发者。

从官网的 .NET SDK 下载量来看，.NET 开发者每年都在快速增长。其中，在新增长的 .NET 开发者中，有 40% 是学生，说明有大量年轻开发者正在涌入 .NET 大家庭中。

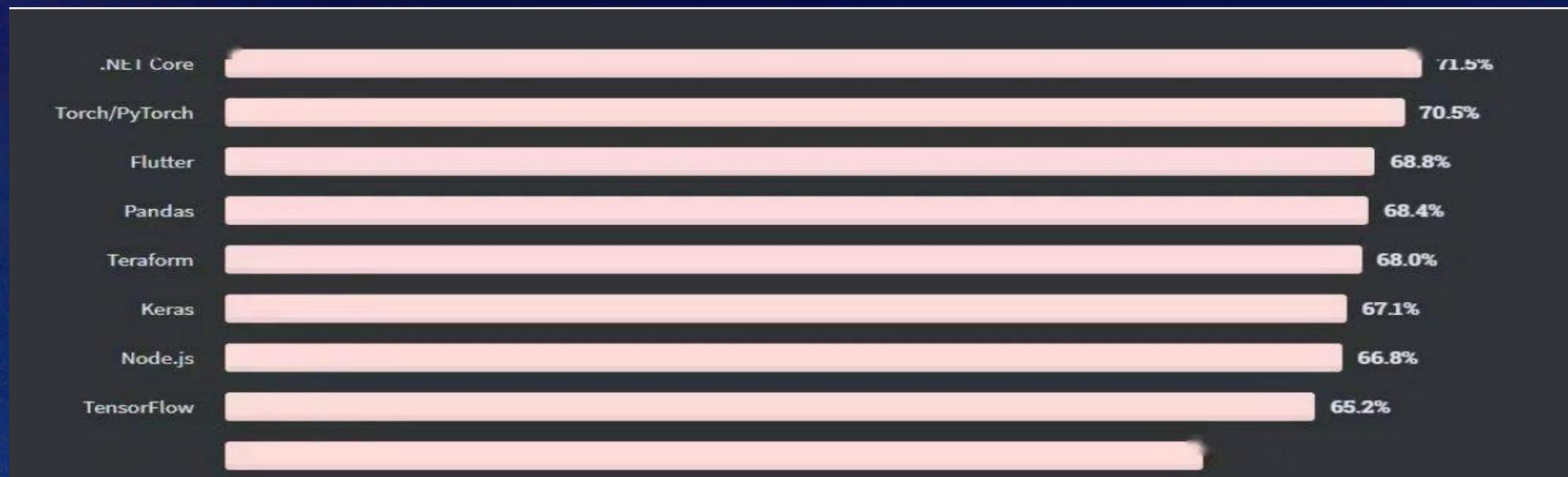


.NET的现状



社区发展

在 2019 年和 2020 年的 Stack Overflow 开发者年度调查报告中，.NET Core 连续两年被评为最受欢迎的开发框架。



.NET的现状



社区发展

.NET 平台下的开源项目，在 GitHub 近两年开源项目贡献(包含 Commit、Issue 等)增长量排名中名列前三十，参与贡献的开发者非常多，全球有 6 万多名开发者贡献了代码。C# 编程语言在 GitHub 的使用量排名中名列前五。



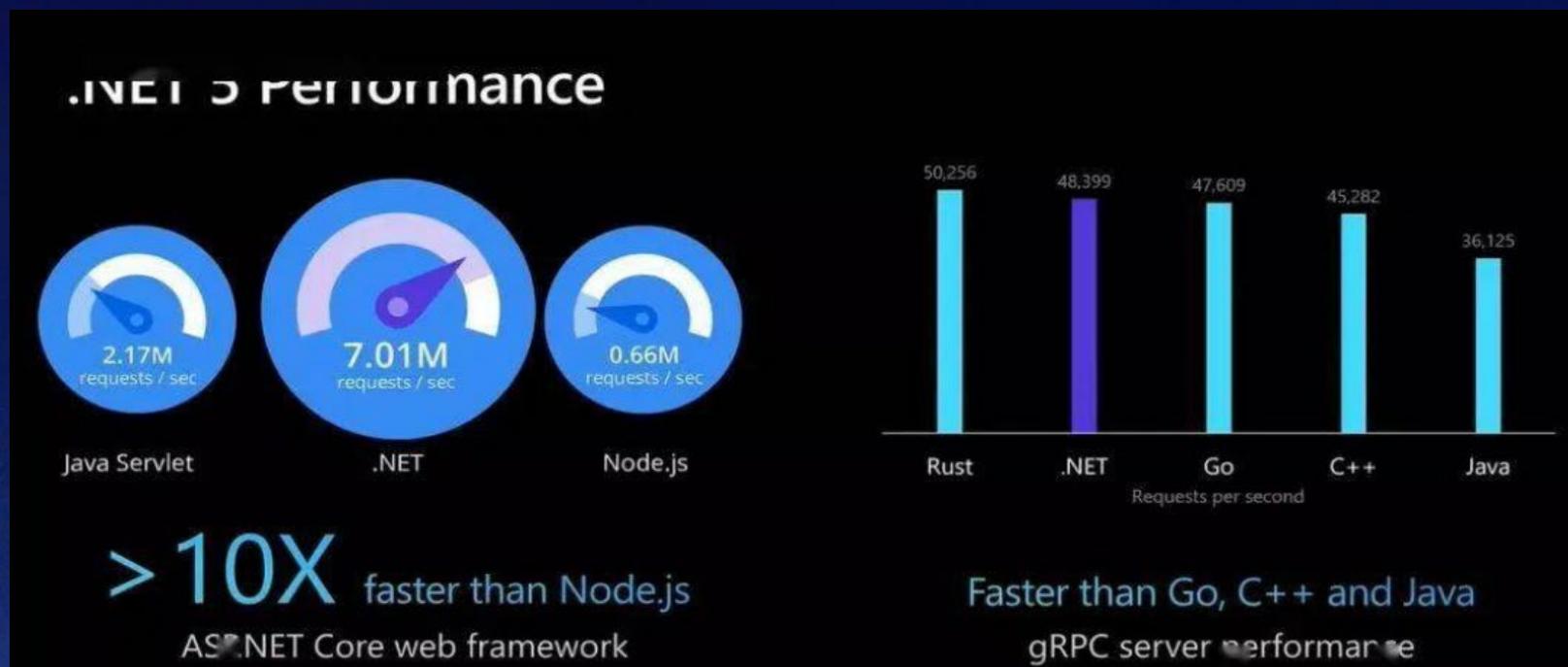
.NET的现状



性能表现

在 .NET 相关框架大版本发布时，几乎每次都有关于性能提升的报告。ASP.NET Core 的性能比 Node.js 快 10 倍多，与 Go 的 Web 框架旗鼓相当。有参考意义的是对 gRPC 实现的框架性能的比较，它比较能反应综合性能的差异。请看下图右边图表，基于 .NET 5 实现的 gRPC 框架与其它 gRPC 框架的对比：

：



.NET的现状



开发者数量

截止 2021年 5 月，全球有超过 500 万的 .NET 开发者。过去两年的跨平台 .NET Core 活跃开发者数量增长超过 60 万。而这些数据仅来源于 Visual Studio IDE，不包含使用其它 IDE 的开发者。

从官网的 .NET SDK 下载量来看，.NET 开发者每年都在快速增长。其中，在新增长的 .NET 开发者中，有 40% 是学生，说明有大量年轻开发者正在涌入 .NET 大家庭中。



.NET的现状

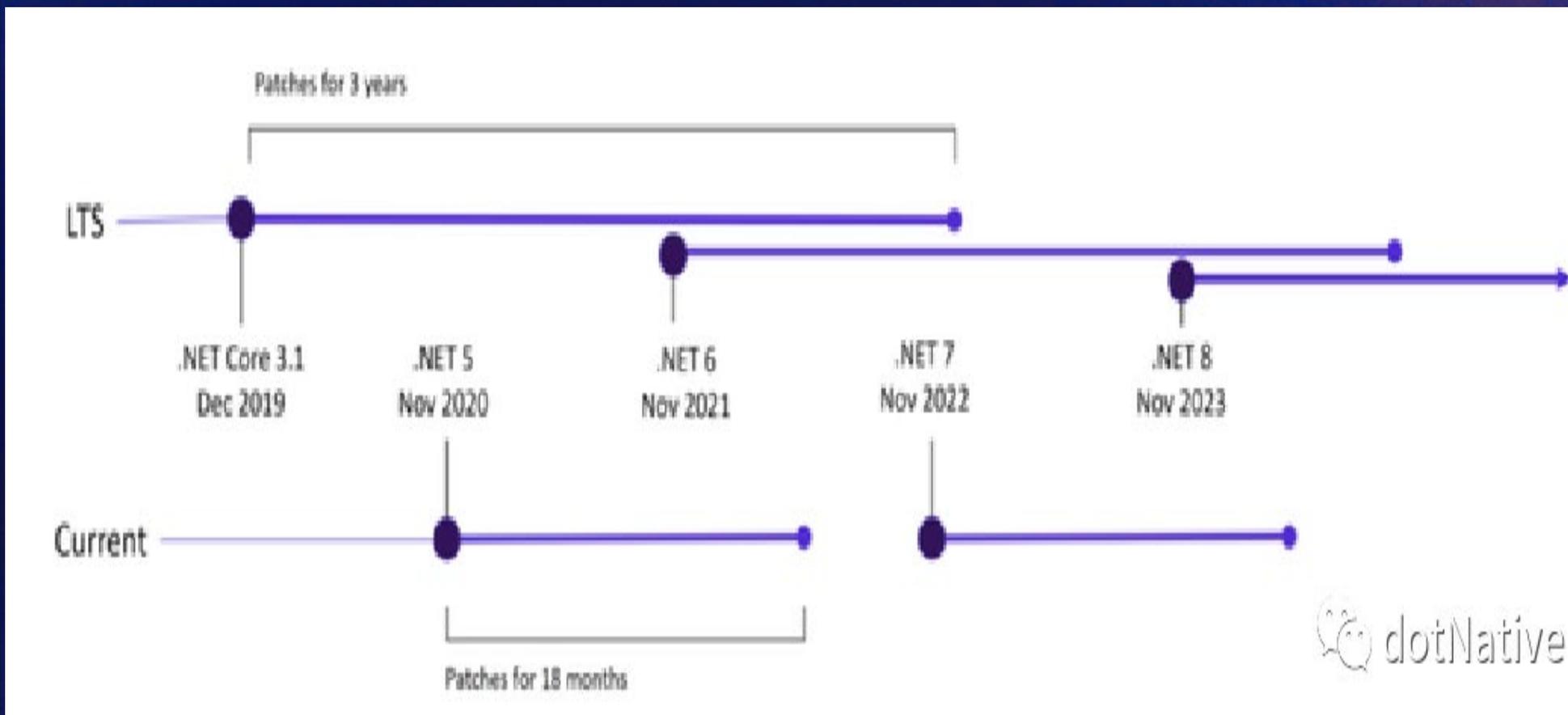


其它方面

.NET 的现状还包括生态发展情况，.NET 是目前全球市场上唯一一个涵盖 Web、桌面客户端、手机应用、微服务、云原生、机器学习、游戏、物联网等跨平台的全能开发解决方案。同时各领域开源的开发框架和库也越来越多。近几年，微软和 .NET 社区做了大量艰辛的工作，使得 .NET 的发展越来越趋向良好，生态也越来越完善。在国内也有很多大厂采用 .NET 技术，比如腾讯、阿里云和网易等，甚至我们的国产 CPU 龙芯也在热情拥抱 .NET。尤其是大型手游开发这块，C# 的使用率独占鳌头，比如腾讯的很多手游都是基于 C# 使用 Unity 开发的，包括著名的王者荣耀。



.NET 的未来



.NET Conf China

.NET Schedule



- .NET Core 3.0 release in September
- .NET Core 3.1 = Long Term Support (LTS)
- .NET 5.0 release in November 2020
- Major releases every year, LTS for even numbered releases
- Predictable schedule, minor releases if needed

 dotNative

.NET 的
未来



.NET Conf China

.NET 20 YEARS

免费，跨平台，开源
用于构建所有应用的开发人员平台。

开始使用

下载

Windows、Linux 和 macOS 上均支持

浏览 .NET



Web

为 Windows、Linux、macOS、
Docker 构建 Web 应用和服务。



手机

使用单一代码库生成适用于 iOS、
Android 和 Windows 等的本机移动
应用。



桌面

创建适用于 Windows 和 macOS 的
本机应用，或使用 Web 技术生成
随时随地运行的应用。



微服务

创建可在 Docker 容器上运行的可
独立部署的微服务。



云

使用现有云服务，或创建和部署自
己的云服务。



机器学习

为应用添加视觉算法、语音处理、
预测模型等。



游戏开发

为最热门的台式机、手机和控制台
开发 2D 和 3D 游戏。



物联网

使用 Raspberry Pi 和其他单板计算
机的本机支持创建 IoT 应用。

语言

从使用 C# 或 Visual Basic 编写 .NET 应用。

详细了解 .NET 语言。



.NET 的未 来

.NET Conf China

我如何学习.NET



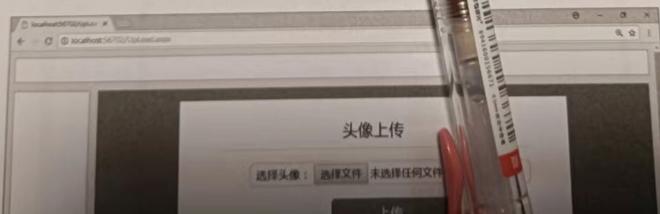
第3章 高级控件

```
color: #F00;  
font-family: '微软雅黑', '文泉驿正黑', '黑体';
```

(4) 找到上传按钮，双击产生单击事件，在单击事件中加入上传图片代码，并且把图片上传到 Images 文件夹。

```
if (FileUpload1.HasFile)  
{  
    // 获取上传的文件名  
    string fileName = FileUpload1.FileName;  
    // 获取上传文件的文件后缀名  
    String fileFix = fileName.Substring(fileName.LastIndexOf('.') + 1).ToLower();  
    if (fileFix != ".png" && fileFix != ".jpg" && fileFix != ".gif" && fileFix != ".pic")  
    {  
        this.lblMessage.Text = "上传的文件不是图片类型！";  
    }  
    else  
    {  
        FileUpload1.SaveAs(Server.MapPath(".") + "~/Images/" + fileName);  
        this.imgUser.ImageUrl = "~/Images/" + fileName;  
        this.lblMessage.Text = "图片上传成功！";  
    }  
}
```

(5) 用鼠标右键选择 Upload.aspx 页面，选择设置为起始页，将该页面设置为默认启动页面。
(6) 运行程序。点击主菜单里“调试”菜单中的“开始调试”或按 F5 来运行，如图 3.14 所示。



④ Operator: 设置比较运算符, 如等于 (Equal), 大于 (Greater Than Equal), 小于等于 (Less Than Equal), 大于 (Greater Than) 和小于 (Less Than) 等。默认设置为等于

⑤ ValueToCompare: 用于比较的值

3. 范围验证控件

RangeValidator 控件用于检查输入的值是否在指定的最小值和最大值之间。

用属性

ControlToValidate: 要验证的控件的 ID

MaximumValue: 范围的上界 (最大值)

MinimumValue: 范围的下界 (最小值)

ValidationType: 验证类型 (字符串、整型、双精度、日期和货币)

4. 自定义验证控件

Validator 控件用于执行自定义验证。

ControlToValidate: 要验证的控件的 ID

ValidationFunction: 设置客户端验证的脚本函数 (事件)

ValidateMethod: 服务器端验证的方法

5. 错误汇总验证控件

ValidationSummary 控件用于在页面上显示所有验证错误的摘要, 它收集本页中所有验证控件的验证信息 (ErrorMessage 属性), 进行组织并统一显示。只能以简单列表、段落或弹出框的形式显示错误信息。

常用属性

① ShowMessageBox: 是否显示弹出的提示消息

② ShowSummary: 是否显示该报告内容



```

class DBHelper
{
    public static string connStr = "server=.;database=
    =test;uid=sa;pwd=123456";
    public static SqlConnection conn = new SqlConnection(connStr);
    // 方法一: 执行查询操作, 返回结果集
    public static DataTable Select(string sql)
    {
        SqlDataAdapter dap = new SqlDataAdapter(sql, conn);
        DataTable dt = new DataTable();
        dap.Fill(dt); // 使用数据适配器来填充的DataTable对象
        return dt;
    }
    // 方法二: 执行非查询操作(增,删,改), 返回执行后是否影响行
    public static bool Update(string sql)
    {
        conn.Open(); // 打开目标数据库;
        SqlCommand cmd = new SqlCommand(sql, conn);
        // 调用ExecuteNonQuery方法, 返回数据库受影响的行数
        int rows = cmd.ExecuteNonQuery();
        conn.Close(); // 关闭数据库连接
        return rows > 0;
    }
}

```

01 初识类和对象

抽象
是指对具体问题概括, 抽取某类对象的公共特征并加以描述的过程, 抽象的过程就是对问题进行分析认识的过程。

类
是对事物的一种抽象定义, 将抽象之后的特征和行为有机地结合起来便构成了类, 类是具有共同特征和行为的一类事物的统称。

C# 类定义
C#语言是一种面向对象的语言, 是编写应用程序的基础, C#中的类主要由成员变量和成员方法组成。

语法如下:

```

[访问修饰符] class 类名 {
    类的字段声明;
    类的属性声明;
    类的方法声明;
}

```

快捷键: prop + tab + tab {get;set}

程序分为以下四个步骤:
1. 建立连接
2. 发送请求, 生成HTTP响应并接收服务器端的响应。

02 验证控件

1. 非空验证控件

↓
RequiredFieldValidator 控件要求用户在页面上输入必填的内容, 常用于文本框“不能为空”的验证

常用属性

- ControlToValidate: 要验证的控件的ID
- Text: 验证出错时的提示
- ErrorMessage: 提交给汇总控件的错误信息, 如果Text属性为空, 则验证出错, 将显示该信息

2. 比较验证控件

↓
CompareValidator 控件用于执行数据类型检查, 比较一个值和另一个值, 以确定这两个值是否与此算符指定的关系相匹配。

常用属性

- ControlToValidate: 要验证的控件的ID
- ControlToCompare: 用于与要验证的控件进行比较的控件
- Type: 设置比较类型(如字符串, 整型等)的比较会出错





- 第1章 ASP.NET Web应用程序基础
 - ▶ 1.1 Web应用程序简介 (5.75min) >
 - ▶ 1.2 HTTP协议 (7.23min) >
 - ▶ 1.3 ASP.NET简介与Web Forms创建 (9.92min) >
 - ▶ 1.4 ASP.NET基本控件 (7.23min) >
- 第2章 验证控件
 - ▶ 2.1 数据验证控件 (9.48min) >
 - ▶ 2.2 非空和比较验证控件 (6.10min) >
 - ▶ 2.3 范围和正则表达式验证控件 (4.93min) >
 - ▶ 2.4 自定义验证控件 (11.48min) >
- 第3章 高级控件
 - ▶ **3.1 母版页与内容页 (10.90min) >**
 - ▶ 3.2 FileUpload文件上传控件 (12.62min) >
 - ▶ 3.3 文件下载 (9.60min) >
 - ▶ 3.4 TreeView控件 (14.60min) >
- 第4章 系统对象
 - ▶ 4.1 Page对象 (11.47min) >
 - ▶ 4.2 Response对象 (7.20min) >
 - ▶ 4.3 Request对象 (14.73min) >
 - ▶ 4.4 Server对象 (9.53min) >

知识要点

母版页与内容页：

母版页主要有以下四个优点：

- (1) 有利于站点修改和维护，降低开发人员的工作强度。
- (2) 具备高效的内容整合能力。



.NET知识 —C#



C#—知识



C# 是一个现代的、通用的、面向对象的编程语言，它是由微软（Microsoft）开发的，由 Ecma 和 ISO 核准认可的。

C# 是由 Anders Hejlsberg 和他的团队在 .Net 框架开发期间开发的。

C# 是专为公共语言基础结构（CLI）设计的。CLI 由可执行代码和运行时环境组成，允许在不同的计算机平台和体系结构上使用各种高级语言。

C# 成为一种广泛应用的专业语言的原因：

1. 现代的、通用的编程语言。
2. 面向对象。
3. 面向组件。
4. 容易学习。
5. 结构化语言。
6. 它产生高效率的程序。
7. 它可以在多种计算机平台上编译。
8. .Net 框架的一部分。



理解C#程序结构



- 在C#中，程序源文件以.cs作为扩展名
- using关键字，用于导入命名空间，” using system “语句导入system命名空间，所以system命名空间中的类和方法可以直接使用
- namespace关键字用于声明命名空间、组织和管理代码，把一些紧密相关的代码放在同一个命名空间中，提高管理和使用效率
- class关键字用于声明类，C#是完全面向对象的语言，类是组织代码的最小单位，编写的代码都应该在一个类里面。
- Main () 方法被称为主方法，是所有C#程序的入口，任何C#程序均以Main()方法开始运行，并在Main () 方法中结束运行，Main () 中的M需要大写，Main方法返回值可以是void类型或者int类型，Main () 括号内可以没有参数
- 注释：单行注释为 //，多行注释为以/*开头 ， 以 */结尾
- 在C#中标点符号都是英文，不然会出错



Visual Studio



Visual Studio (VS) 是目前最专业，最流行的Windows平台应用程序开发环境。

VS的开发界面由五大部分组成：

- 菜单和工具栏：各种功能的快捷方式
- 工具箱区：很多的控件
- 工作区：编写代码
- 解决方案区：用于管理项目
- 输出区：显示项目编译后或者运行后的相关信息



下次

再

遇

Thank you!

Let's build amazing apps with .NET 7
get.dot.net/7

