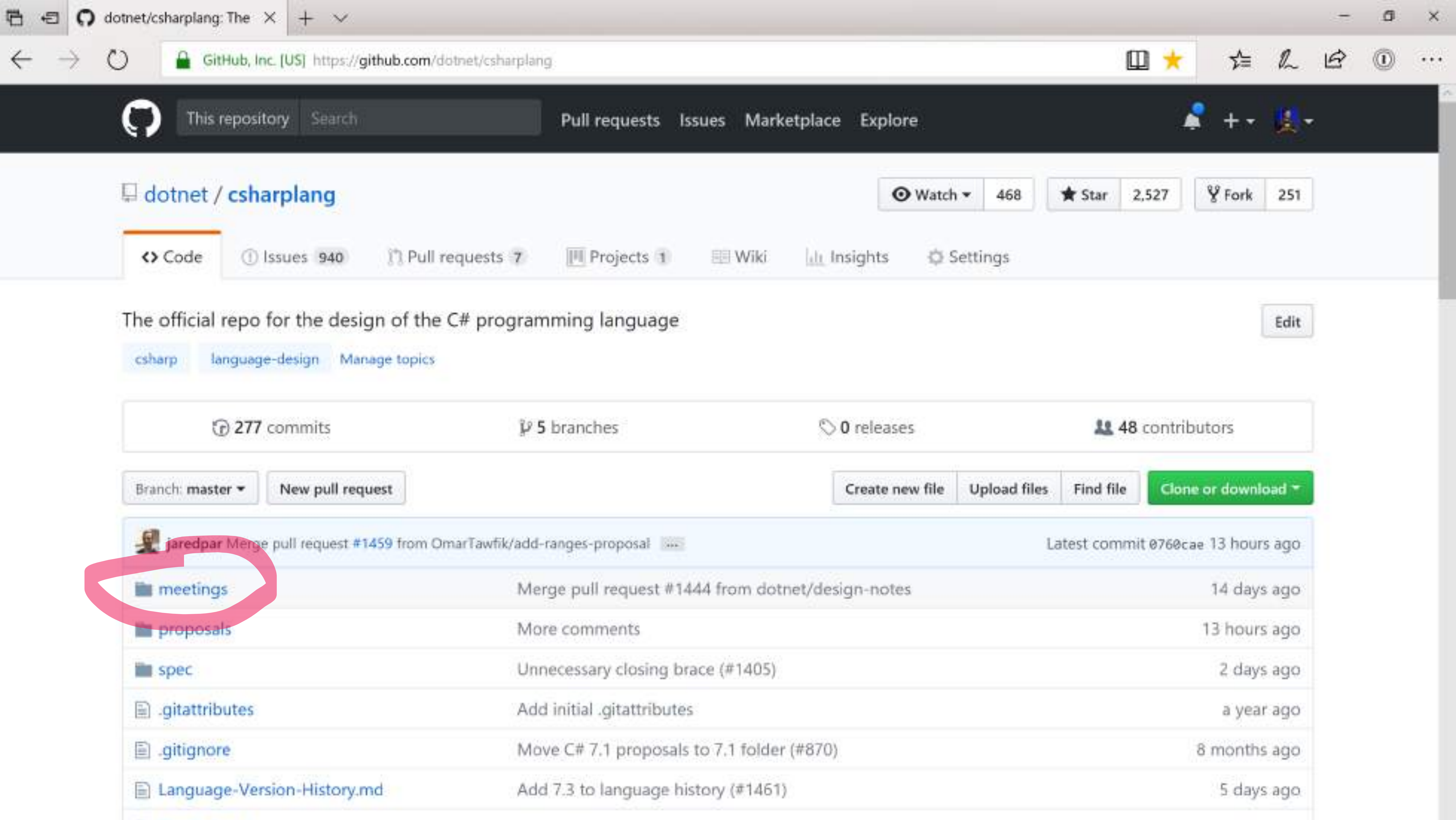


dotnet / csharplang

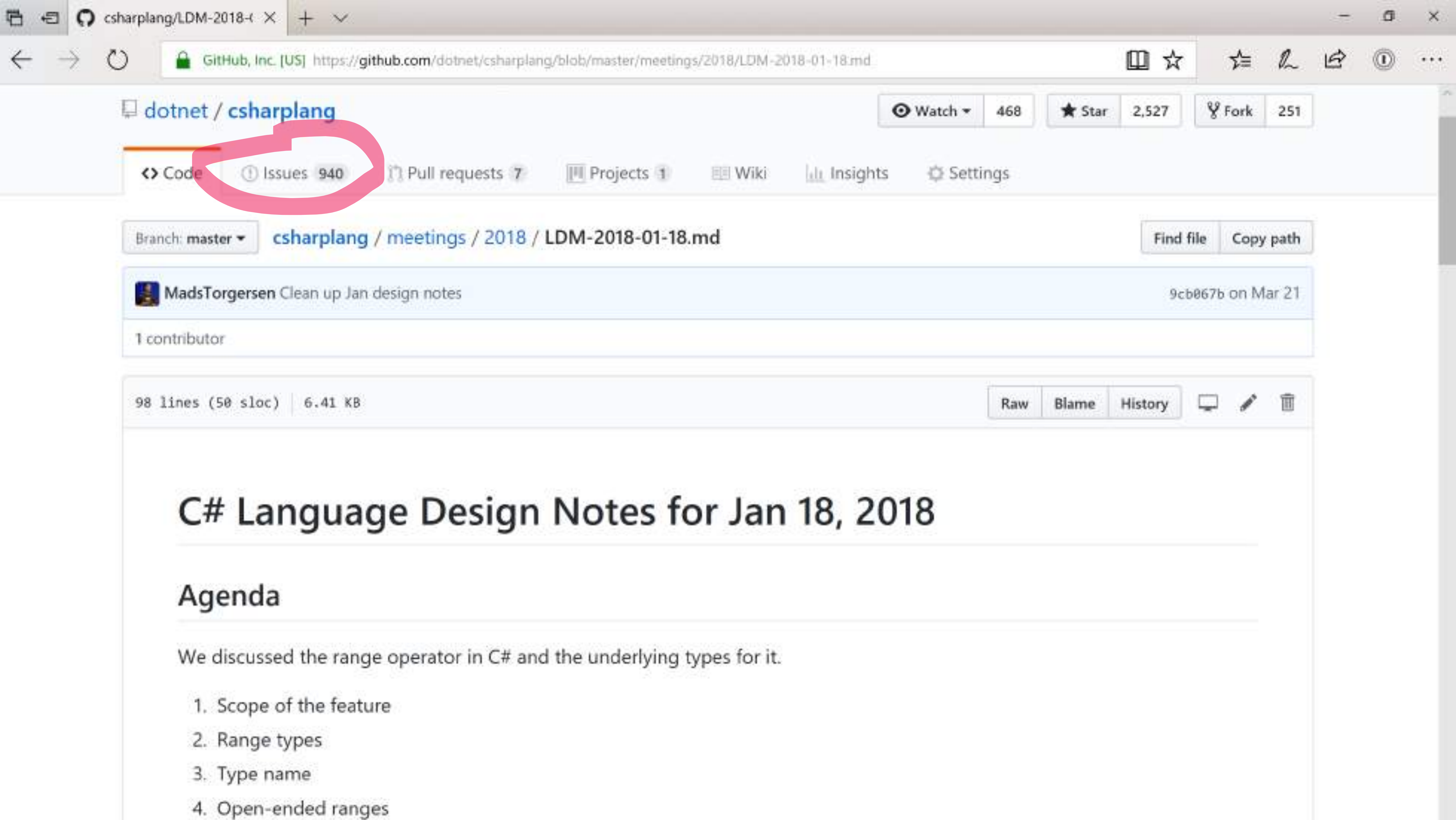
The official repo for the design of the C# programming language

csharp language-design

277 commits 5 branches 0 releases 48 contributors

jaredpar Merge pull request #1459 from OmarTawfik/add-ranges-proposal … Latest commit 0760cae 13 hours ago

meetings	Merge pull request #1444 from dotnet/design-notes	14 days ago
proposals	More comments	13 hours ago
spec	Unnecessary closing brace (#1405)	2 days ago
.gitattributes	Add initial .gitattributes	a year ago
.gitignore	Move C# 7.1 proposals to 7.1 folder (#870)	8 months ago
Language-Version-History.md	Add 7.3 to language history (#1461)	5 days ago

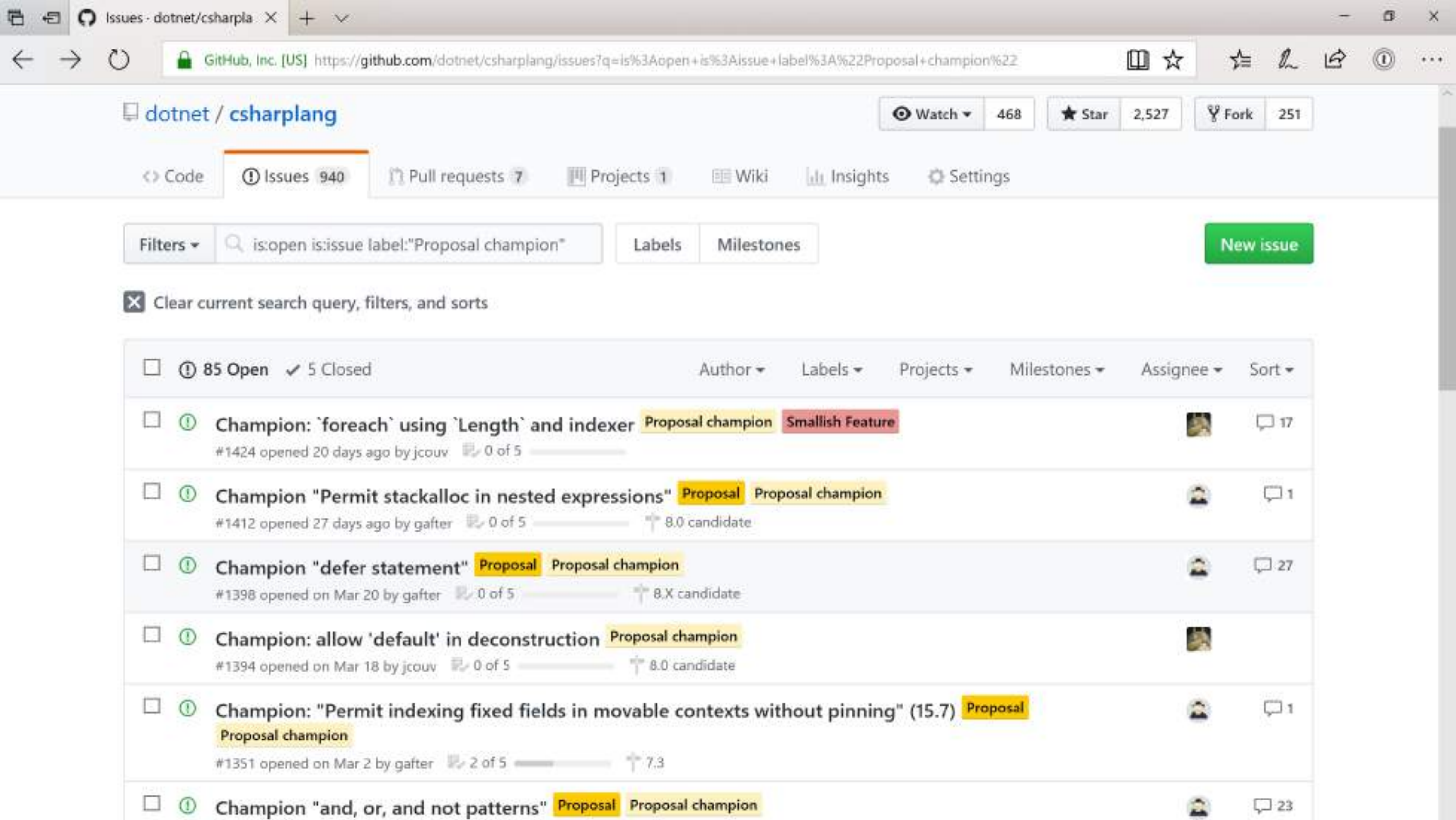

csharplang/LDM-2018-(
GitHub, Inc. [US] https://github.com/dotnet/csharplang/blob/master/meetings/2018/LDM-2018-01-18.md
dotnet / csharplang
Watch 468
Star 2,527
Fork 251
<> Code
Issues 940
Pull requests 7
Projects 1
Wiki
Insights
Settings
Branch: master
csharplang / meetings / 2018 / LDM-2018-01-18.md
Find file
Copy path
MadsTorgersen Clean up Jan design notes
9cb067b on Mar 21
1 contributor
98 lines (50 sloc) | 6.41 KB
Raw
Blame
History

C# Language Design Notes for Jan 18, 2018

Agenda

We discussed the range operator in C# and the underlying types for it.

1. Scope of the feature
2. Range types
3. Type name
4. Open-ended ranges



is:open is:issue label:"Proposal champion"

☒ Clear current search query, filters, and sorts

□ 85 Open ✓ 5 Closed

□ Champion: `foreach` using `Length` and indexer **Proposal champion** **Smallish Feature** 17
#1424 opened 20 days ago by jcouv 0 of 5

□ Champion "Permit stackalloc in nested expressions" **Proposal** **Proposal champion** 1
#1412 opened 27 days ago by gafter 0 of 5 8.0 candidate

□ Champion "defer statement" **Proposal** **Proposal champion** 27
#1398 opened on Mar 20 by gafter 0 of 5 8.X candidate

□ Champion: allow 'default' in deconstruction **Proposal champion**
#1394 opened on Mar 18 by jcouv 0 of 5 8.0 candidate

□ Champion: "Permit indexing fixed fields in movable contexts without pinning" (15.7) **Proposal** **Proposal champion** 1
#1351 opened on Mar 2 by gafter 2 of 5 7.3

□ Champion "and, or, and not patterns" **Proposal** **Proposal champion** 23

Going open – open to change

Open source

Runtime, framework, compiler

Cross platform

Runs pretty much anywhere

Open language API “Roslyn”

Syntax trees and more

Cross IDE

OmniSharp

Open design

Language design is a conversation

Open to change

Let’s never be afraid to explore new directions

“New direction” on the Great Wall

LINQ – Language Integrated Query

```
from c in customers           customers
where c.City == "Beijing"     .Where(c => c.City == "Beijing")
select c.Name;                .Select(c => c.Name);
```

```
from c in customers           customers
where c.City == "Beijing"     .Where(c => c.City == "Beijing")
from o in c.Orders            .SelectMany(c => c.Orders, (c, o) => new { c, o })
select (c.Name, o.Price);     .Select(p => (p.c.Name, p.o.Price));
```

Async – Keeping C# imperative since 2012

```
void PlaySongs(int[] ids)
{
    foreach (var id in ids)
    {
        try
        {
            var data = service.GetEncodedData(id);
            var song = Decode(data);
            Play(song);
        }
        catch (Exception e) { /* just skip the song */ }
    }
}
```

Async – Keeping C# imperative since 2012

```
Task PlaySongsAsync(int[] ids)
{
    foreach (var id in ids)
    {
        try
        {
            var data = service.GetEncodedData(id);
            var song = Decode(data);
            Play(song);
        }
        catch (Exception e) { /* just skip the song */ }
    }
}
```

Async – Keeping C# imperative since 2012

```
async Task PlaySongsAsync(int[] ids)
{
    foreach (var id in ids)
    {
        try
        {
            var data = service.GetEncodedData(id);
            var song = Decode(data);
            Play(song);
        }
        catch (Exception e) { /* just skip the song */ }
    }
}
```

Async – Keeping C# imperative since 2012

```
async Task PlaySongsAsync(int[] ids)
{
    foreach (var id in ids)
    {
        try
        {
            var data = await service.GetEncodedDataAsync(id);
            var song = Decode(data);
            Play(song);
        }
        catch (Exception e) { /* just skip the song */ }
    }
}
```

Async – Keeping C# imperative since 2012

```
async Task PlaySongsAsync(int[] ids)
{
    foreach (var id in ids)
    {
        try
        {
            var data = await service.GetEncodedDataAsync(id);
            var song = Decode(data);
            await PlayAsync(song);
        }
        catch (Exception e) { /* just skip the song */ }
    }
}
```

Async – Keeping C# imperative since 2012

```csharp
async Task PlaySongsAsync(int[] ids)
{
    foreach (var id in ids)
    {
        try
        {
            var data = await service.GetEncodedDataAsync(id);
            var song = await Task.Run(() => Decode(data));
            await PlayAsync(song);
        }
        catch (Exception e) { /* just skip the song */ }
    }
}
```

Async – Keeping C# imperative since 2012

```
async Task PlaySongsAsync(int[] ids)
{
    Task currentlyPlaying = Task.FromResult(true);
    foreach (var id in ids)
    {
        try
        {
            var data = await service.GetEncodedDataAsync(id);
            var song = await Task.Run(() => Decode(data));
            await PlayAsync(song);
        }
        catch (Exception e) { /* just skip the song */ }
    }
}
```

Async – Keeping C# imperative since 2012

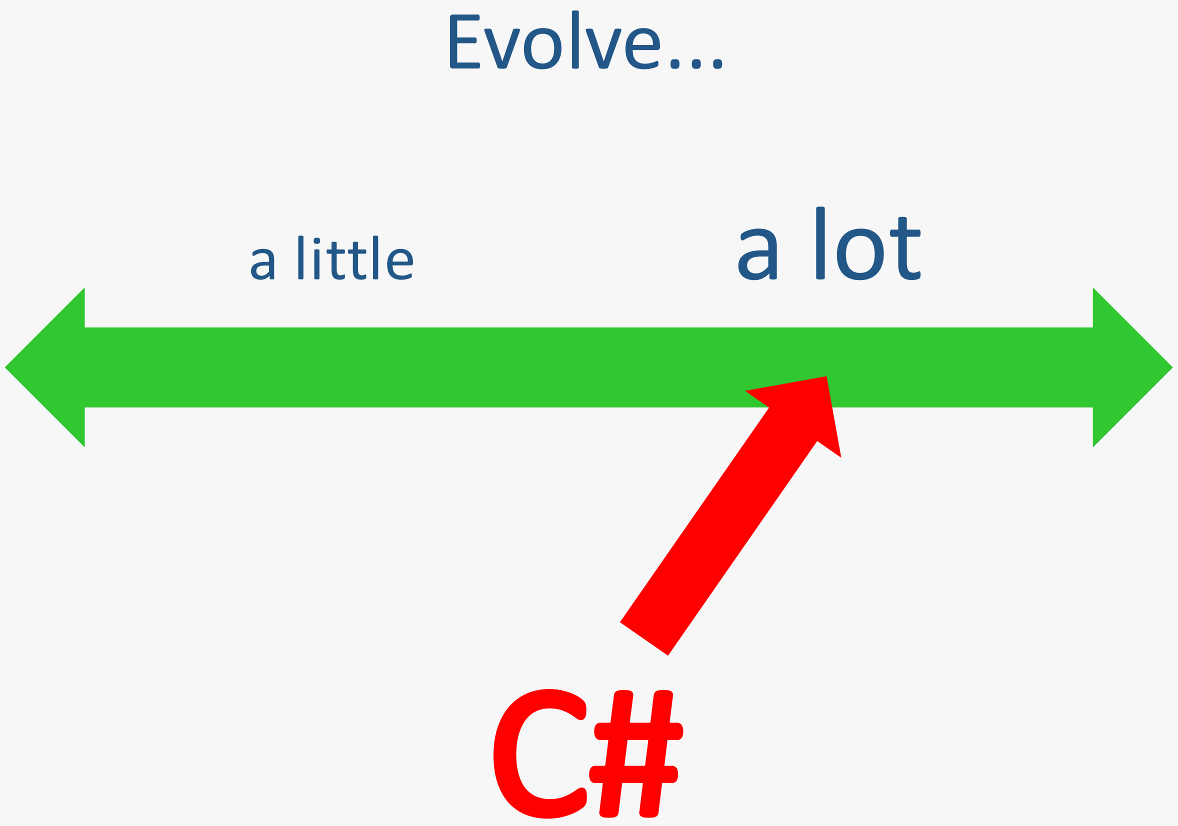
```
async Task PlaySongsAsync(int[] ids)
{
    Task currentlyPlaying = Task.FromResult(true);
    foreach (var id in ids)
    {
        try
        {
            var data = await service.GetEncodedDataAsync(id);
            var song = await Task.Run(() => Decode(data));
            await currentlyPlaying;
            currentlyPlaying = PlayAsync(song);
        }
        catch (Exception e) { /* just skip the song */ }
    }
}
```

Async – Keeping C# imperative since 2012

```
async Task PlaySongsAsync(int[] ids)
{
    Task currentlyPlaying = Task.FromResult(true);
    foreach (var id in ids)
    {
        try
        {
            var data = await service.GetEncodedDataAsync(id);
            var song = await Task.Run(() => Decode(data));
            await currentlyPlaying;
            currentlyPlaying = PlayAsync(song);
        }
        catch (Exception e) { /* just skip the song */ }
    }
    try { await currentlyPlaying; } catch (Exception e) { }
}
```

C# evolution – pace

Stagnation



Evolve...

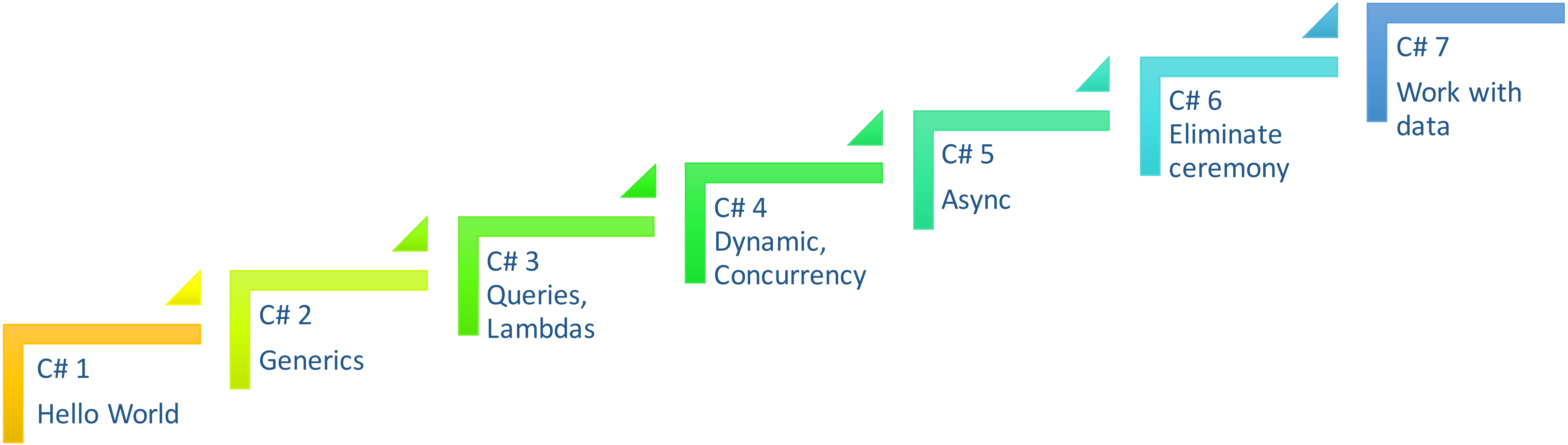
a little

a lot

C#

Explosion

C# evolution – history



C# 1
Hello World

C# 2
Generics

C# 3
Queries,
Lambdas

C# 4
Dynamic,
Concurrency

C# 5
Async

C# 6
Eliminate
ceremony

C# 7
Work with
data

Open for business

Open source

Runtime, framework, compiler

Cross platform

Runs pretty much anywhere

Open language API “Roslyn”

Syntax trees and more

Cross IDE

OmniSharp

Open design

Language design is a conversation

Open mind

Let’s never be afraid to explore new directions

East gate to Beihai Park

主办方 Geekbang 极客邦科技 InfoQ

GMTC2018

全球大前端技术大会

大前端的下一站

<<扫码了解更多详情>>

ArchSummit
全球架构师峰会

关注 ArchSummit 公众号

获取国内外一线架构设计

了解上千名知名架构师的实践动向

Apple • Google • Microsoft • Facebook • Amazon 腾讯 • 阿里 • 百度 • 京东 • 小米 • 网易 • 微博

深圳站：2018年7月6-9日 北京站：2018年12月7-10日

主办方 Geekbang> 极客邦科技 InfoQ

QCon 上海站

全球软件开发大会【2018】

2018年10月18–20日

7折 预售中，现在报名立减2040元

团购享更多优惠，截至2018年7月1日

极客邦科技
企业培训与咨询

Geekbang>.

扫码关注

获取更多培训信息