

# Safari Extensibility: Content Blocking and Shared Links

Session 511

Brian Weinstein Safari and WebKit Engineer

Alex Christensen Safari and WebKit Engineer

# Safari Extensibility

What's new

NEW

Content Blockers

Changes to the Extensions Gallery

Shared Links App Extensions

# Safari Extensibility

What's new

NEW

Content Blockers

Changes to the Extensions Gallery

Shared Links App Extensions

# Safari Extensibility

What's new

NEW

Content Blockers

Changes to the Extensions Gallery

Shared Links App Extensions

# Safari Extensibility

What's new

NEW

Content Blockers

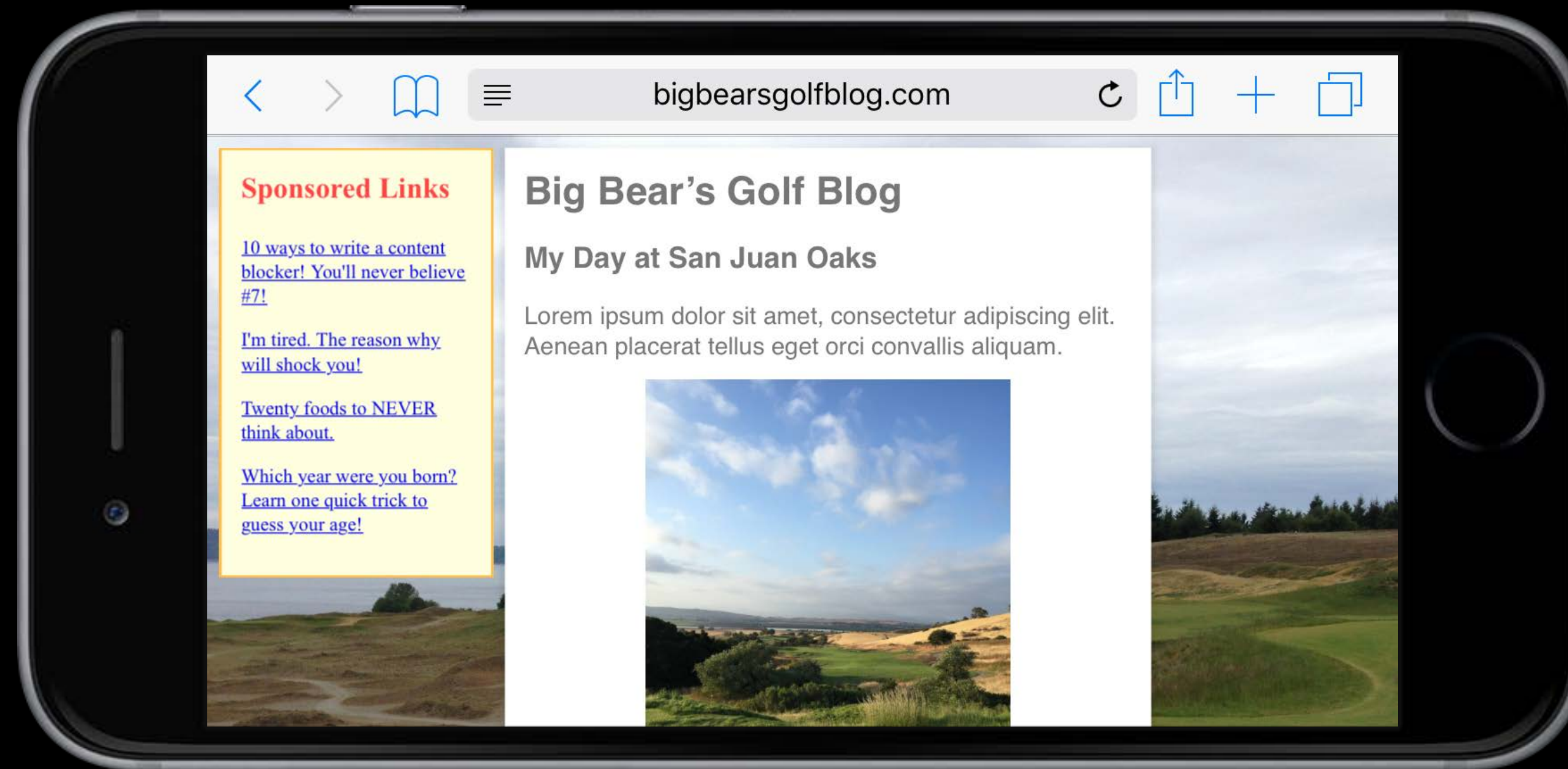
Changes to the Extensions Gallery

Shared Links App Extensions

# Content Blockers

iOS and OS X

# What Content Blockers Can Do





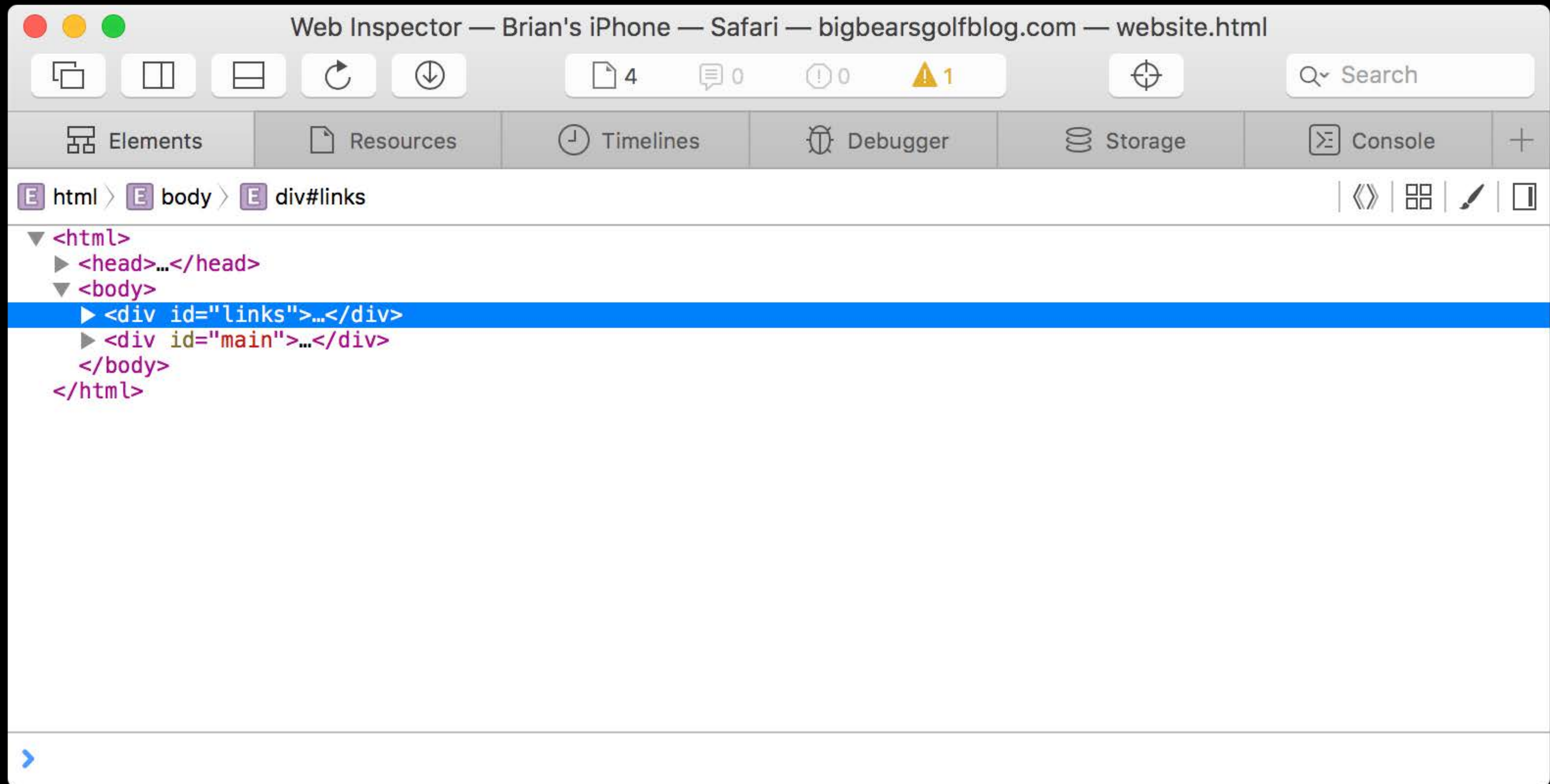
# What Content Blockers Can Do





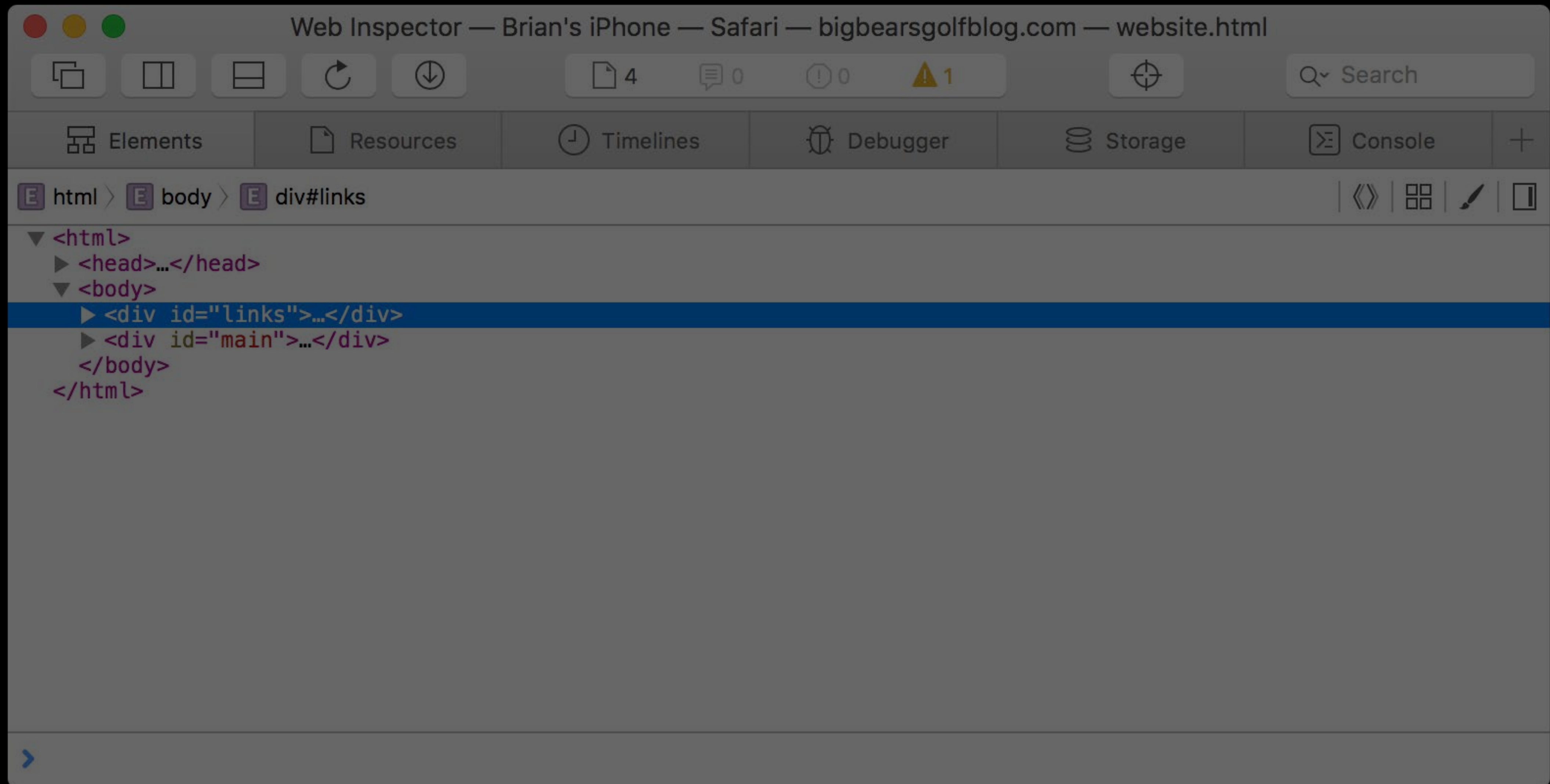
# What Content Blockers Can Do

Hide an element



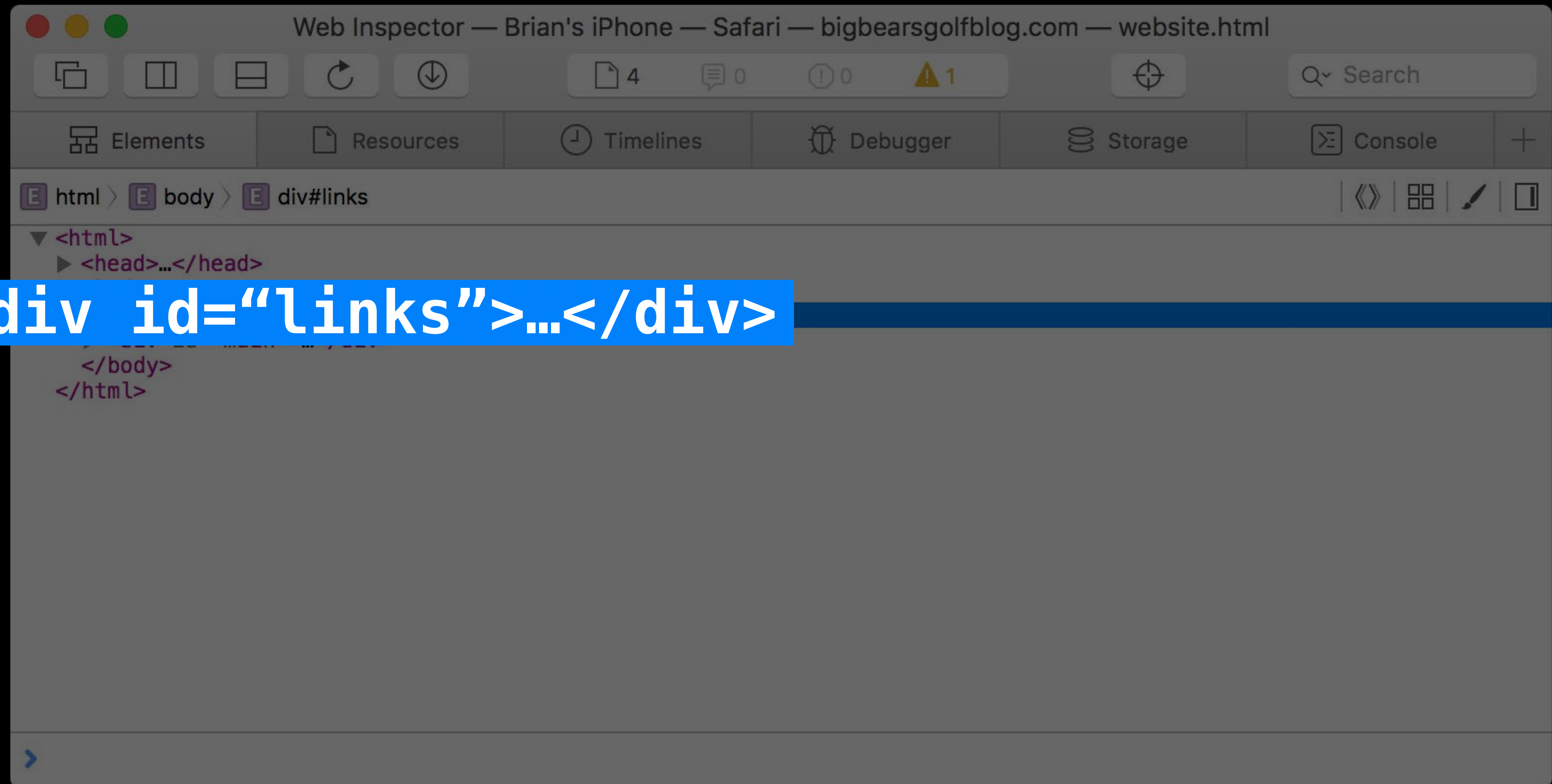
# What Content Blockers Can Do

Hide an element



# What Content Blockers Can Do

Hide an element

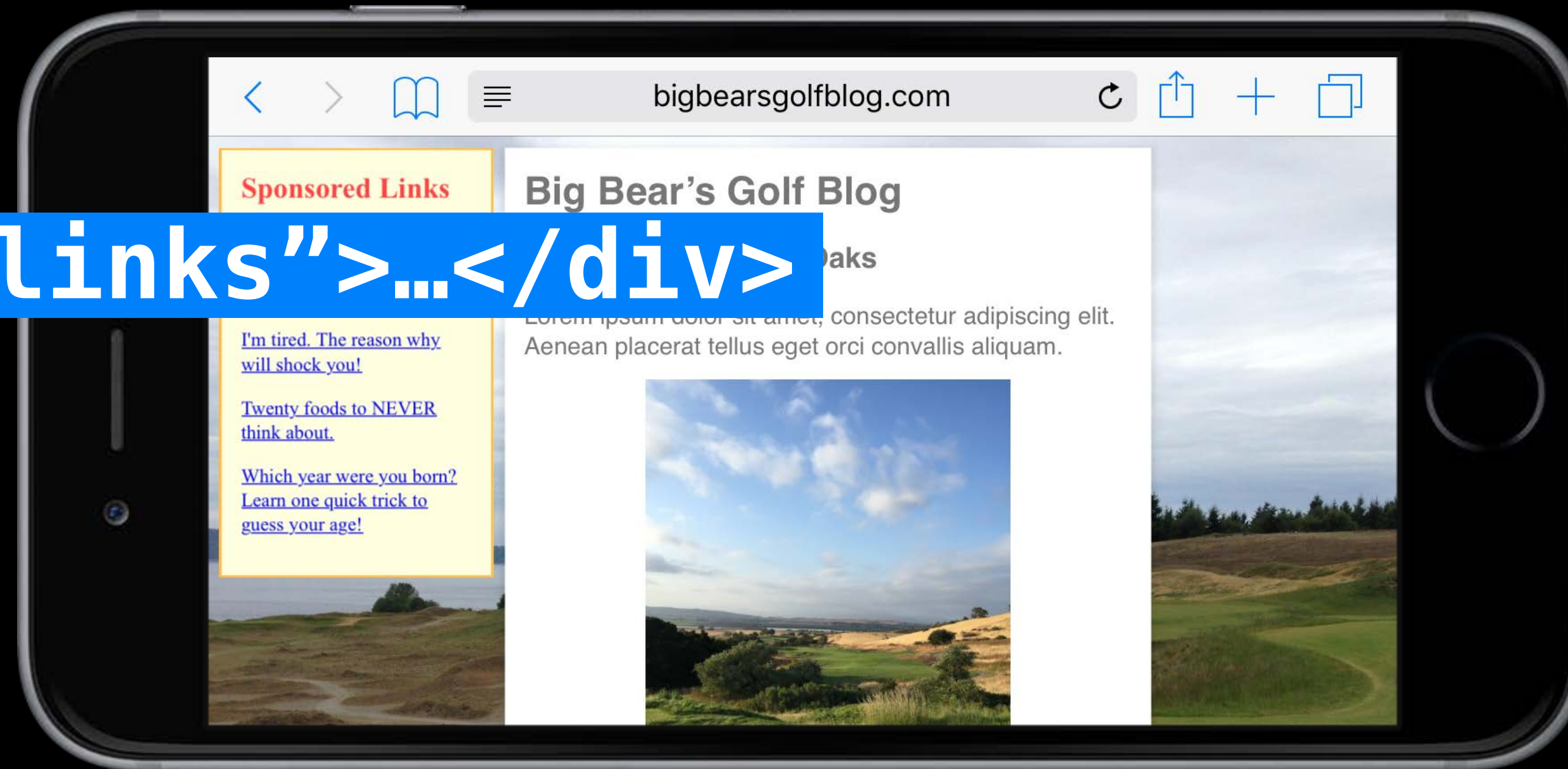




# What Content Blockers Can Do

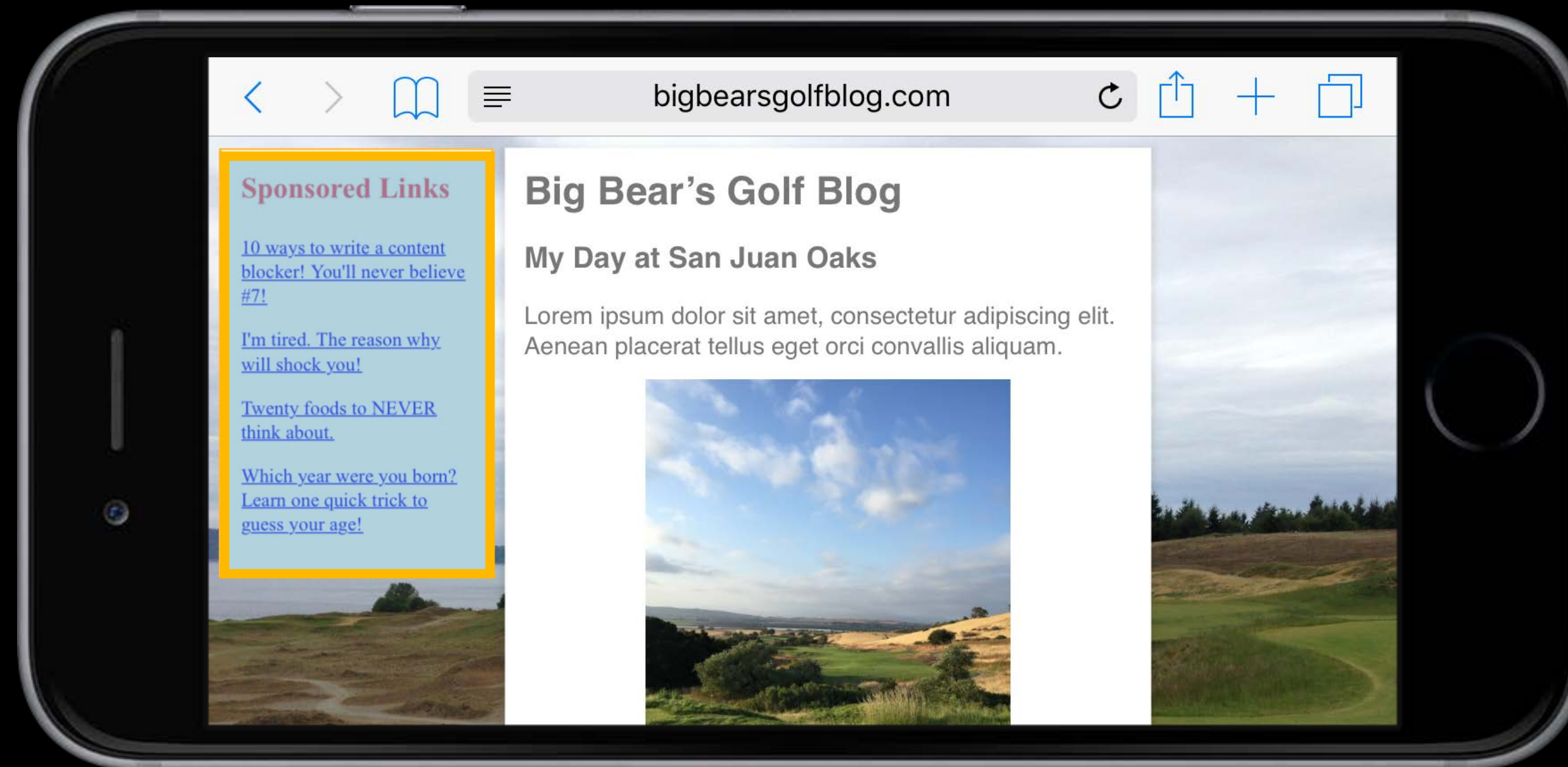
Hide an element

▶ `<div id="links">...</div>`



# What Content Blockers Can Do

Hide an element



# What Content Blockers Can Do

Hide an element

```
"action": {  
  "type": "css-display-none",  
  "selector": "#links"  
}
```

# What Content Blockers Can Do

Hide an element

```
"action": {  
  "type": "css-display-none",  
  "selector": "#links"  
}
```



# What Content Blockers Can Do

Hide an element

```
"action": {  
  "type": "css-display-none",  
  "selector": "#links"  
}
```

# What Content Blockers Can Do

Hide an element

```
"trigger": {  
  "if-domain": ["bigbearsgolfblog.com"],  
  "url-filter": ".*"  
}
```

# What Content Blockers Can Do

Hide an element

```
"trigger": {  
  "if-domain": ["bigbearsgolfblog.com"],  
  "url-filter": ".*"  
}
```

# What Content Blockers Can Do

Hide an element

```
"trigger": {  
  "if-domain": ["bigbearsgolfblog.com"],  
  "url-filter": ".*"  
}
```

# What Content Blockers Can Do

Hide an element

```
{
  "action": {
    "type": "css-display-none",
    "selector": "#links"
  },
  "trigger": {
    "if-domain": ["bigbearsgolfblog.com"],
    "url-filter": ".*"
  }
}
```

# What Content Blockers Can Do

Hide an element

```
{
  "action": {
    "type": "css-display-none",
    "selector": "#links"
  },
  "trigger": {
    "if-domain": ["bigbearsgolfblog.com"],
    "url-filter": ".*"
  }
}
```

# What Content Blockers Can Do

Hide an element

```
{
  "action": {
    "type": "css-display-none",
    "selector": "#links"
  },
  "trigger": {
    "if-domain": ["bigbearsgolfblog.com"],
    "url-filter": ".*"
  }
}
```



# What Content Blockers Can Do

Hide an element

```
{
  "action": {
    "type": "css-display-none",
    "selector": "#links"
  },
  "trigger": {
    "if-domain": ["bigbearsgolfblog.com"],
    "url-filter": ".*"
  }
}
```

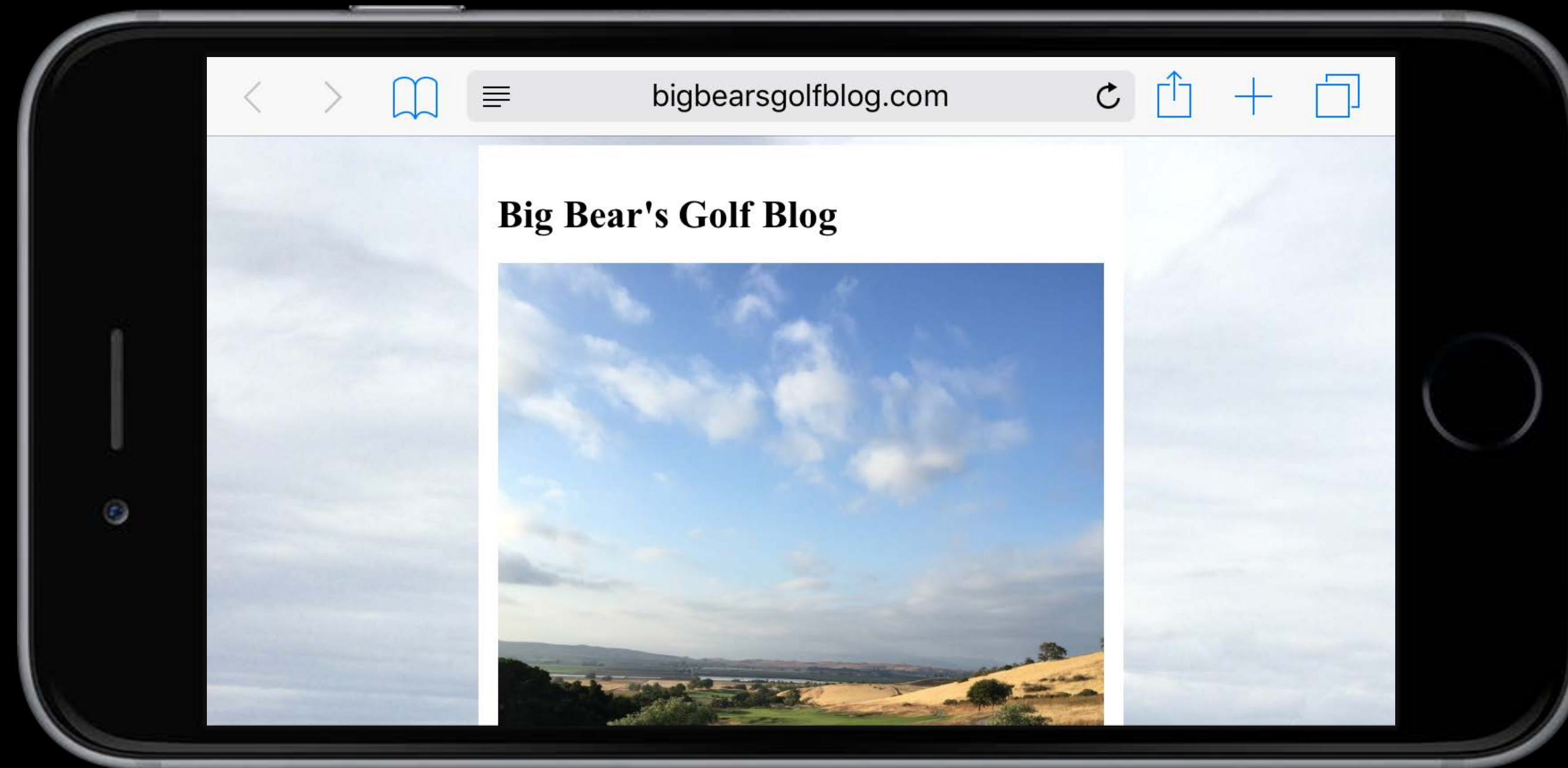
# What Content Blockers Can Do

Hide an element

```
{
  "action": {
    "type": "css-display-none",
    "selector": "#links"
  },
  "trigger": {
    "if-domain": ["bigbearsgolfblog.com"],
    "url-filter": ".*"
  }
}
```

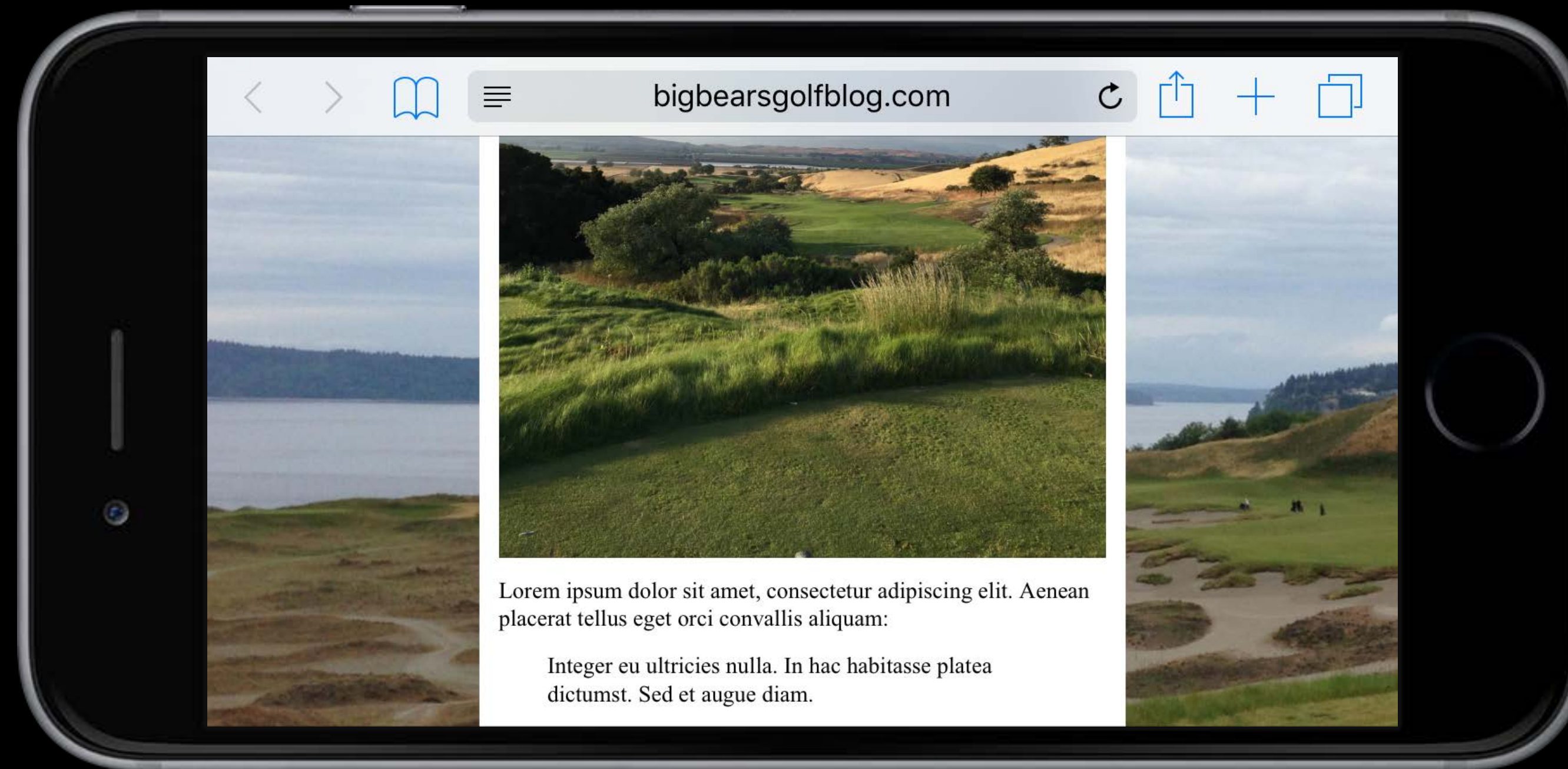
# What Content Blockers Can Do

Hide an element



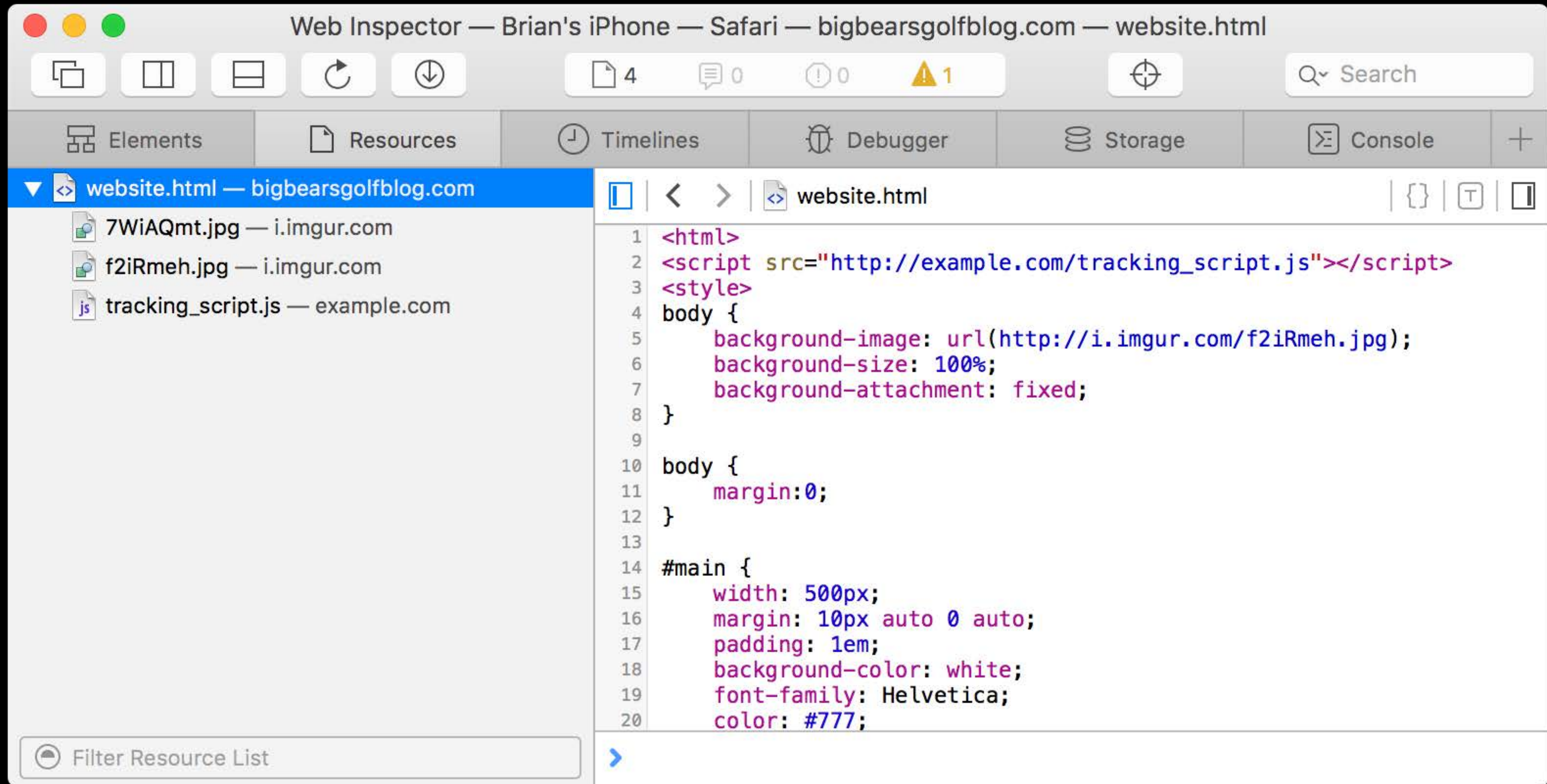
# What Content Blockers Can Do

Hide an element

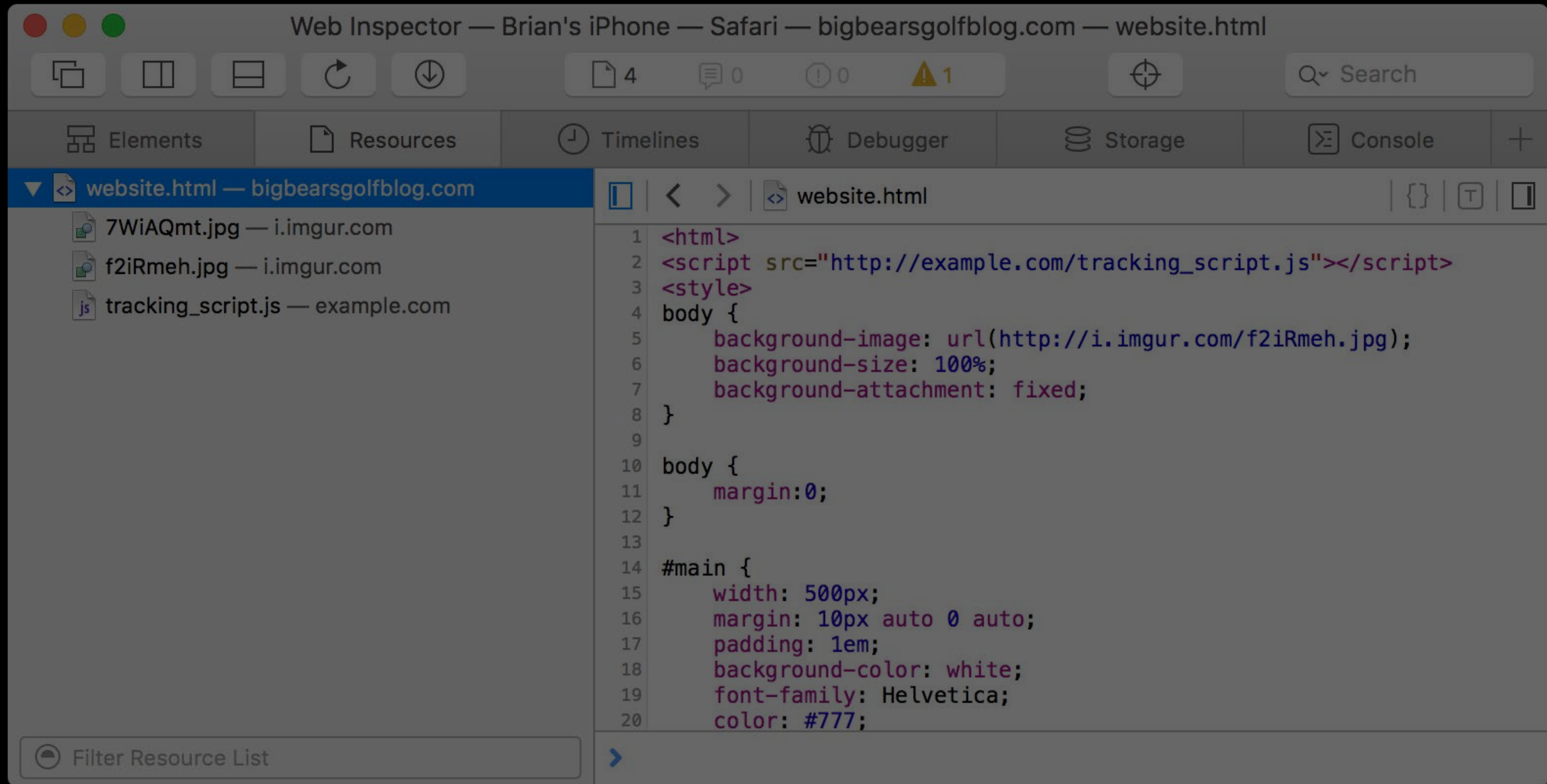




# What Content Blockers Can Do

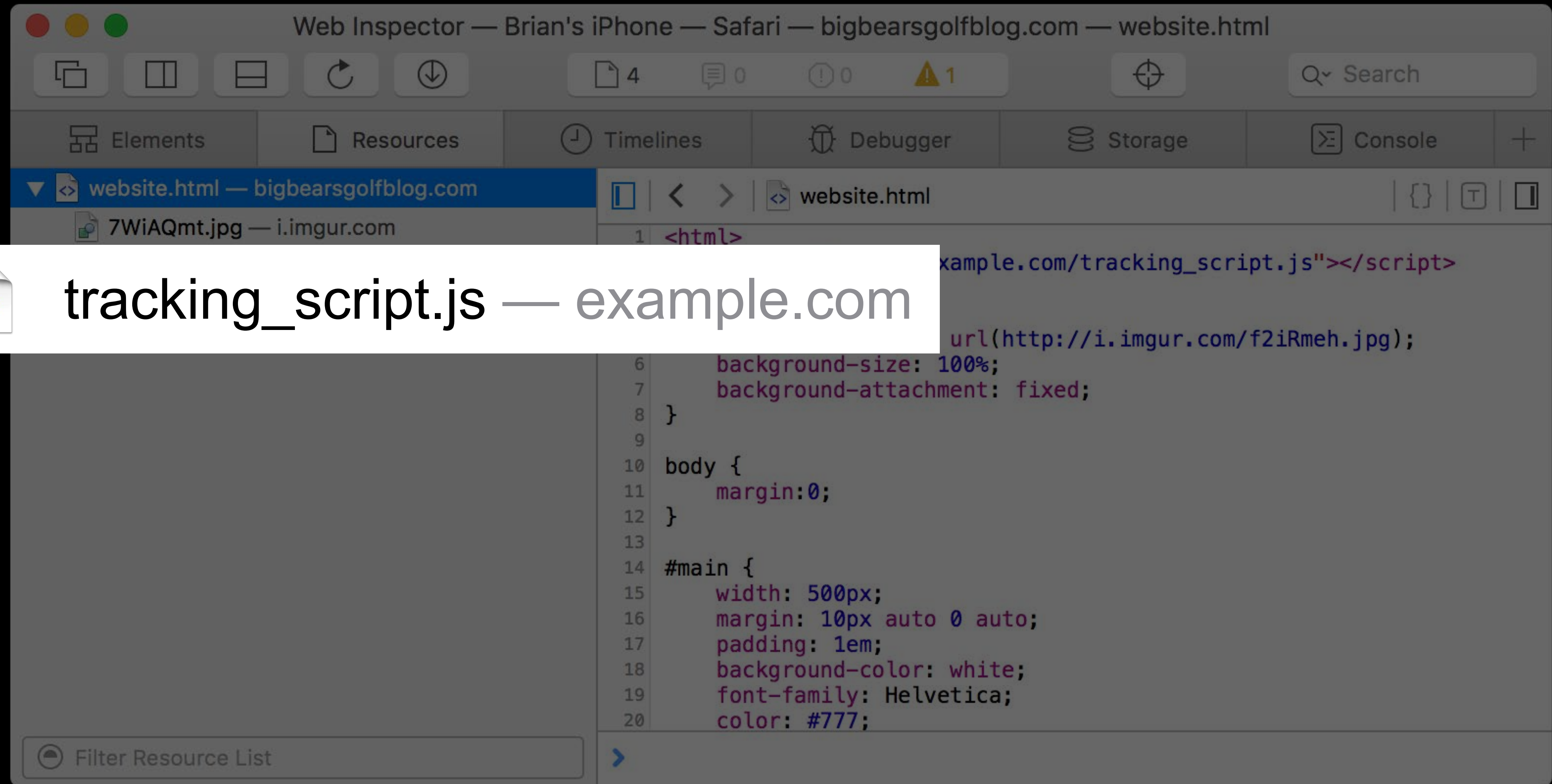


# What Content Blockers Can Do



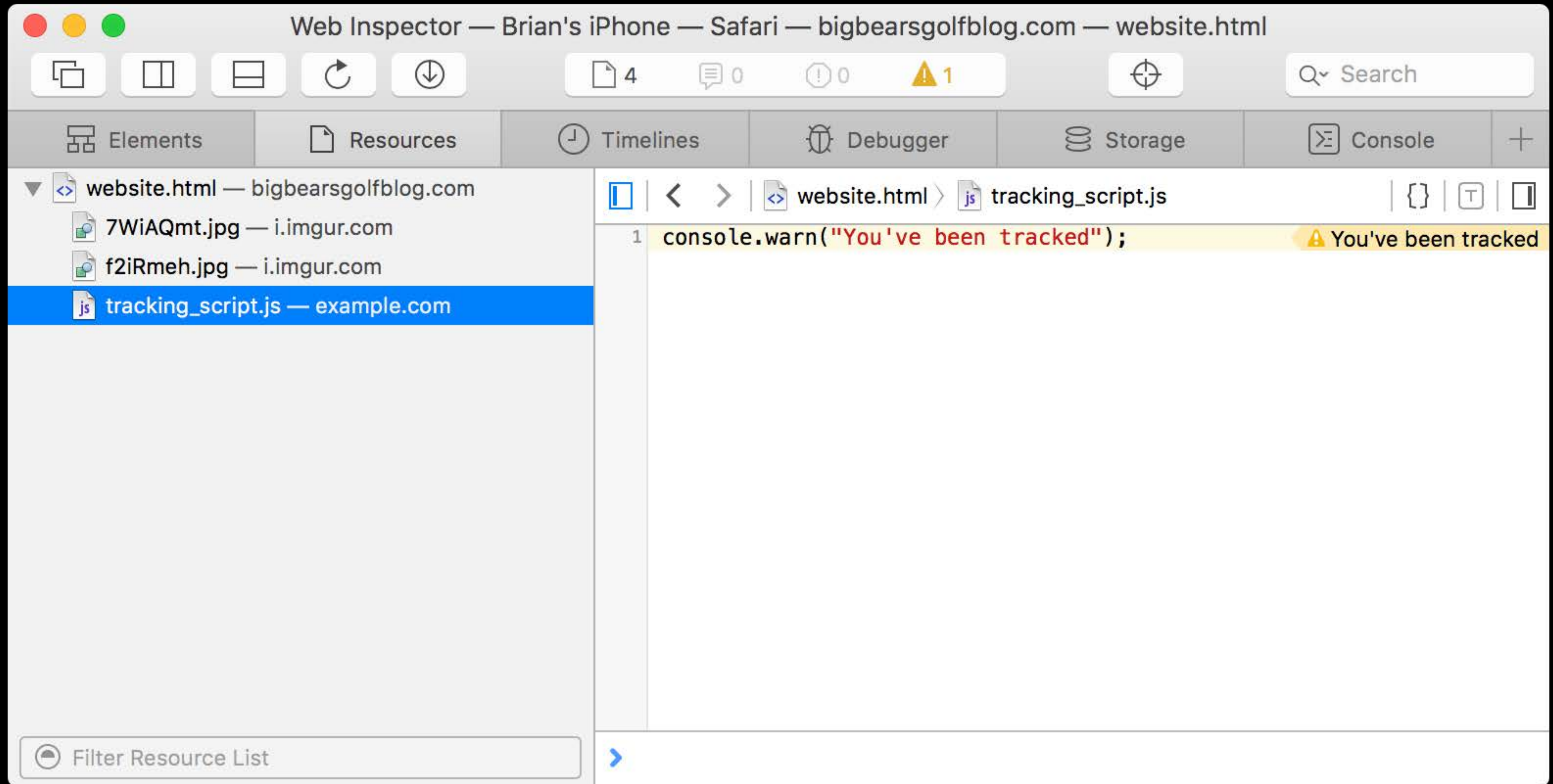


# What Content Blockers Can Do

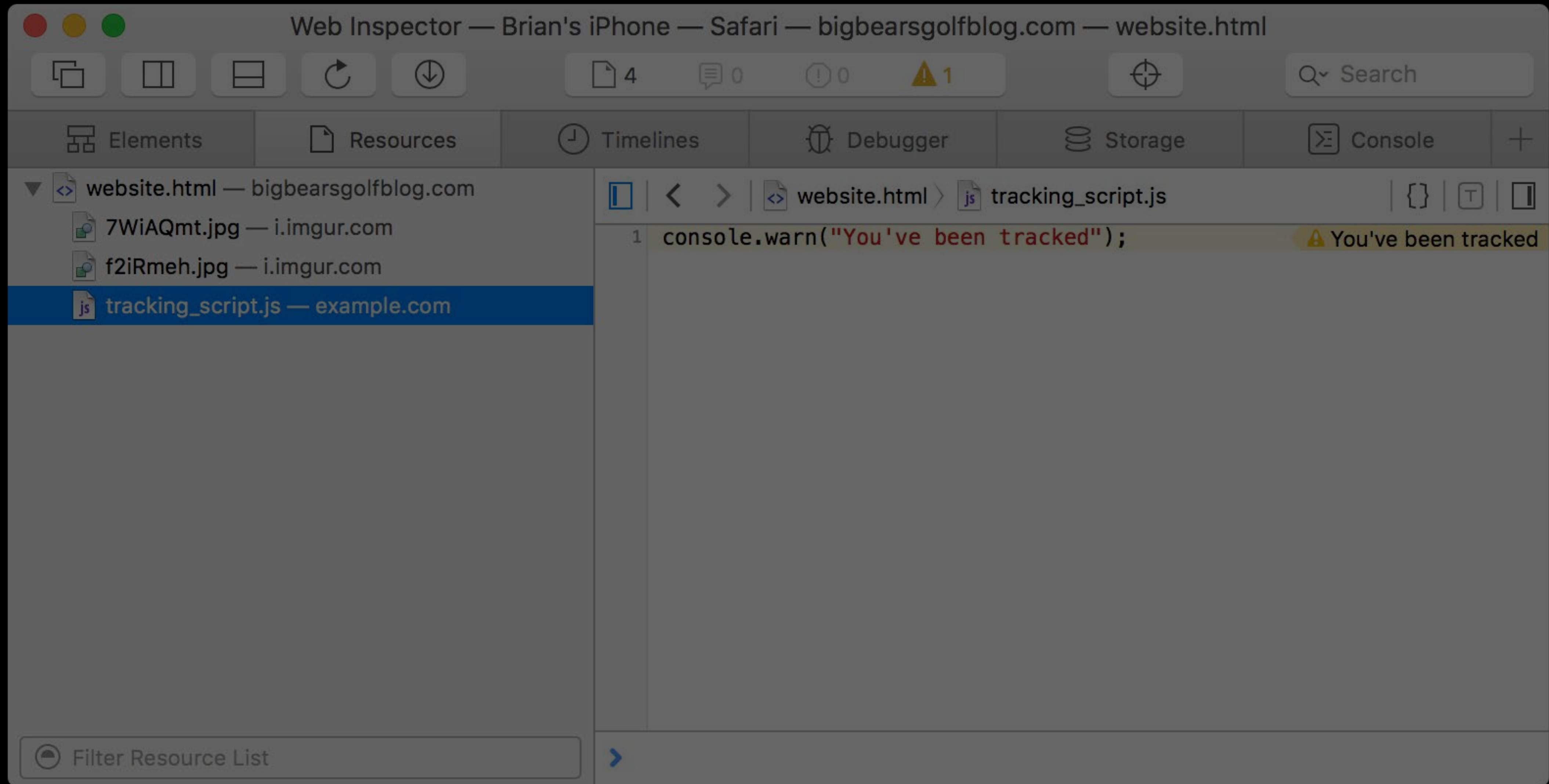




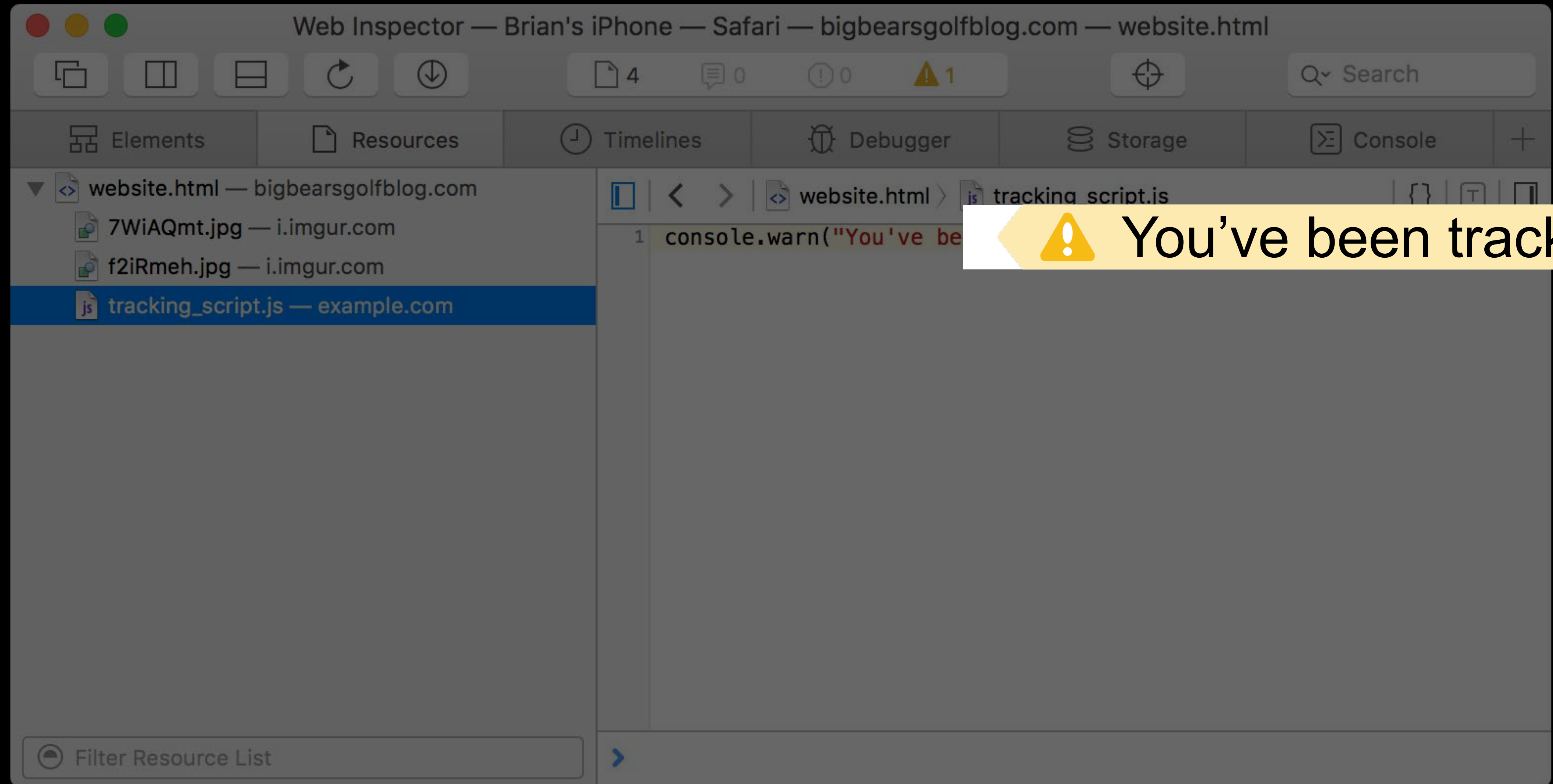
# What Content Blockers Can Do



# What Content Blockers Can Do



# What Content Blockers Can Do



# What Content Blockers Can Do

Block a load

```
"action": {  
  "type": "block"  
}
```

# What Content Blockers Can Do

Block a load

```
"action": {  
  "type": "block"  
}
```



# What Content Blockers Can Do

Block a load

```
"trigger": {  
  "url-filter": "tracking_script",  
  "resource-type": ["script"],  
  "load-type": ["third-party"]  
}
```

# What Content Blockers Can Do

Block a load

```
"trigger": {  
  "url-filter": "tracking_script",  
  "resource-type": ["script"],  
  "load-type": ["third-party"]  
}
```



# What Content Blockers Can Do

Block a load

```
"trigger": {  
  "url-filter": "tracking_script",  
  "resource-type": ["script"],  
  "load-type": ["third-party"]  
}
```

# What Content Blockers Can Do

Block a load

```
"trigger": {  
  "url-filter": "tracking_script",  
  "resource-type": ["script"],  
  "load-type": ["third-party"]  
}
```

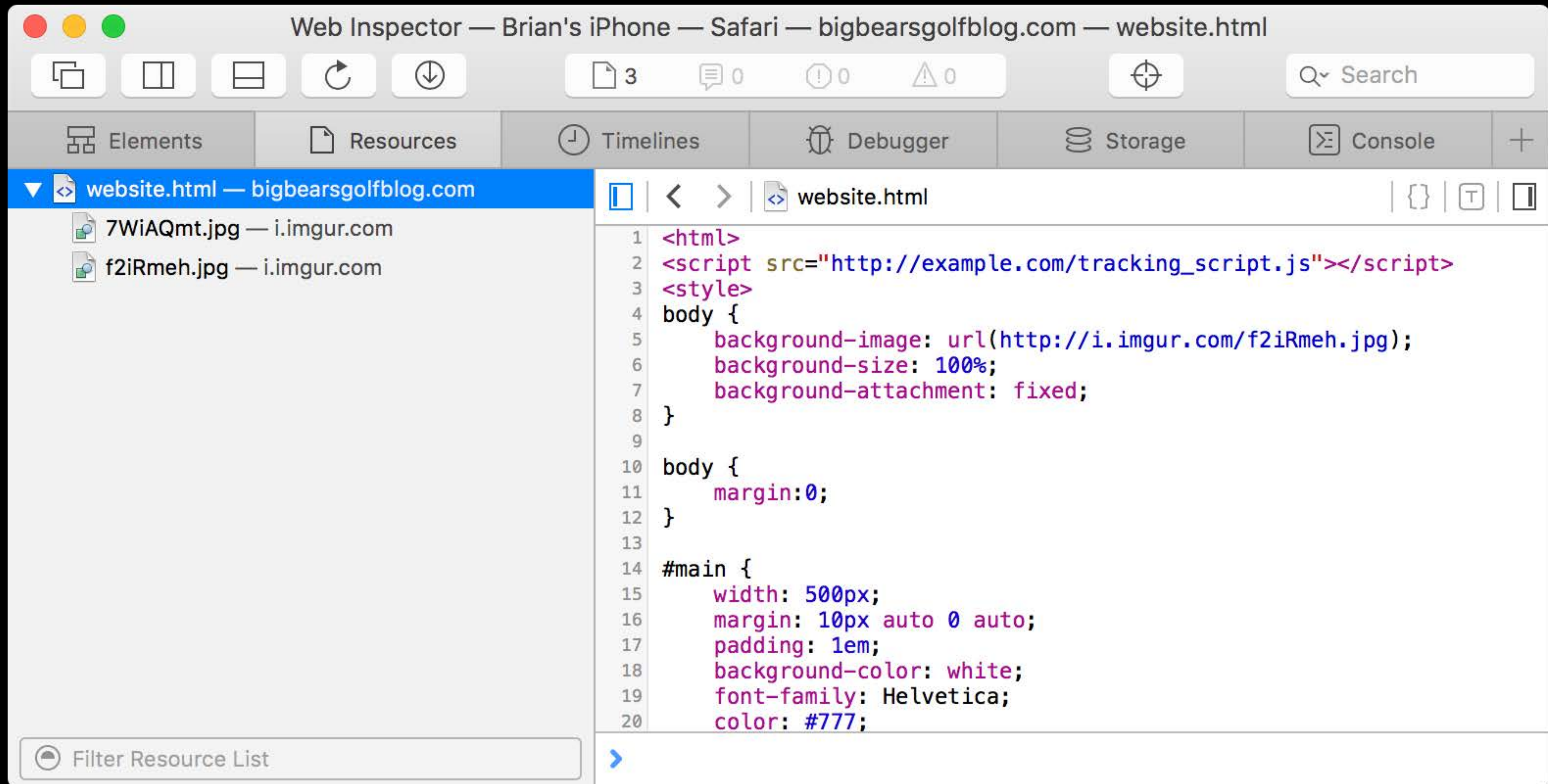
# What Content Blockers Can Do

Block a load

```
{
  "action": {
    "type": "block"
  },
  "trigger": {
    "url-filter": "tracking_script",
    "resource-type": ["script"],
    "load-type": ["third-party"]
  }
}
```

# What Content Blockers Can Do

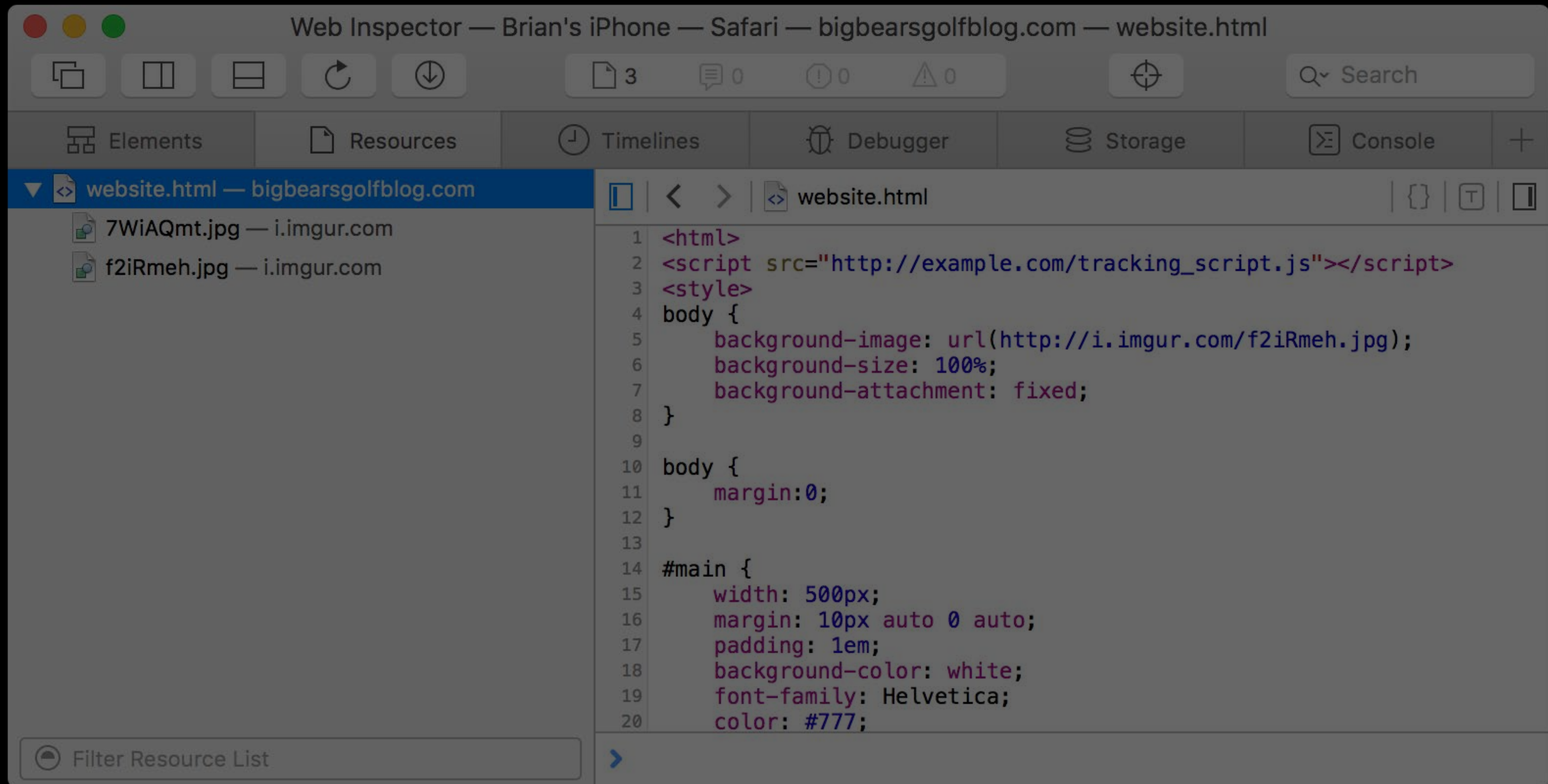
Block a load





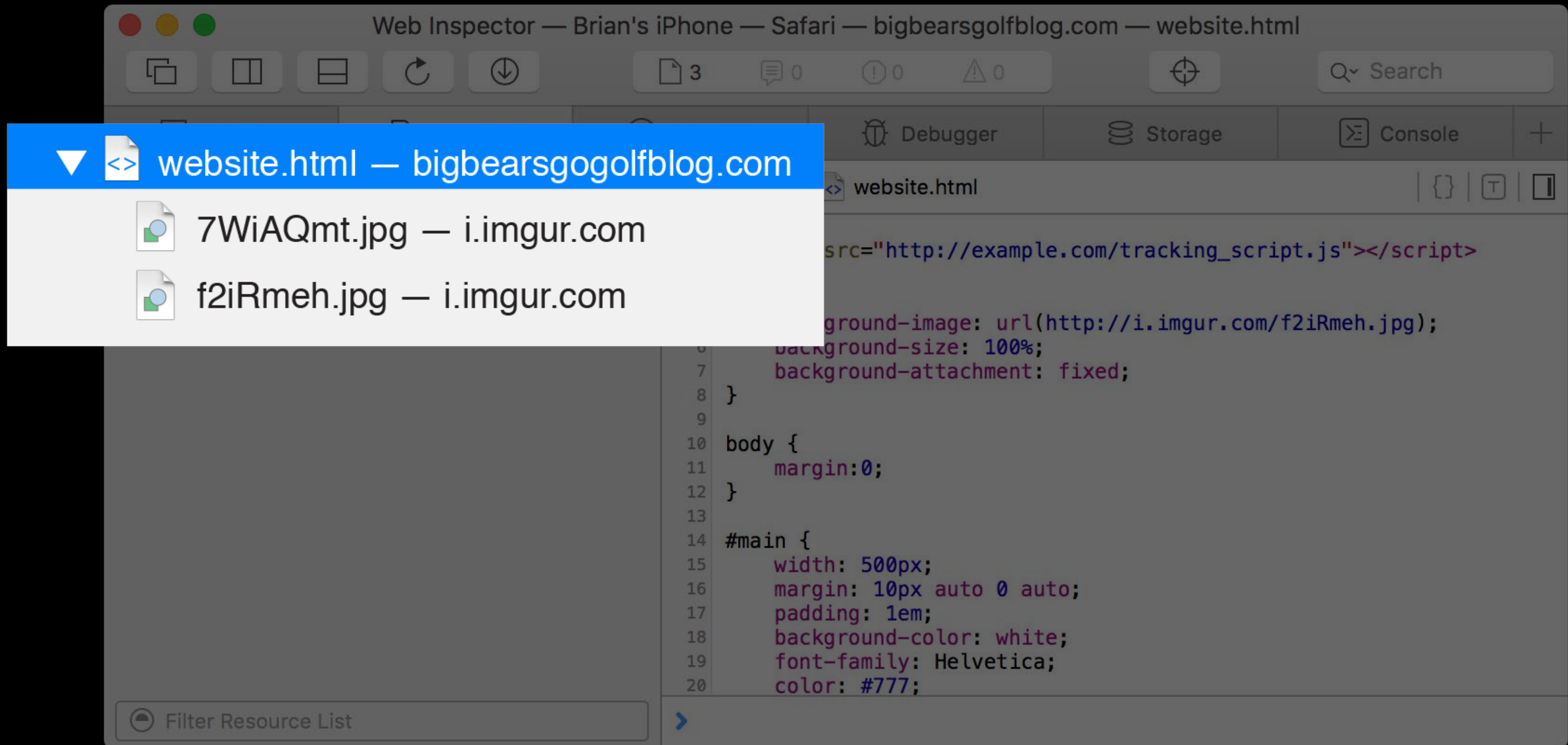
# What Content Blockers Can Do

Block a load



# What Content Blockers Can Do

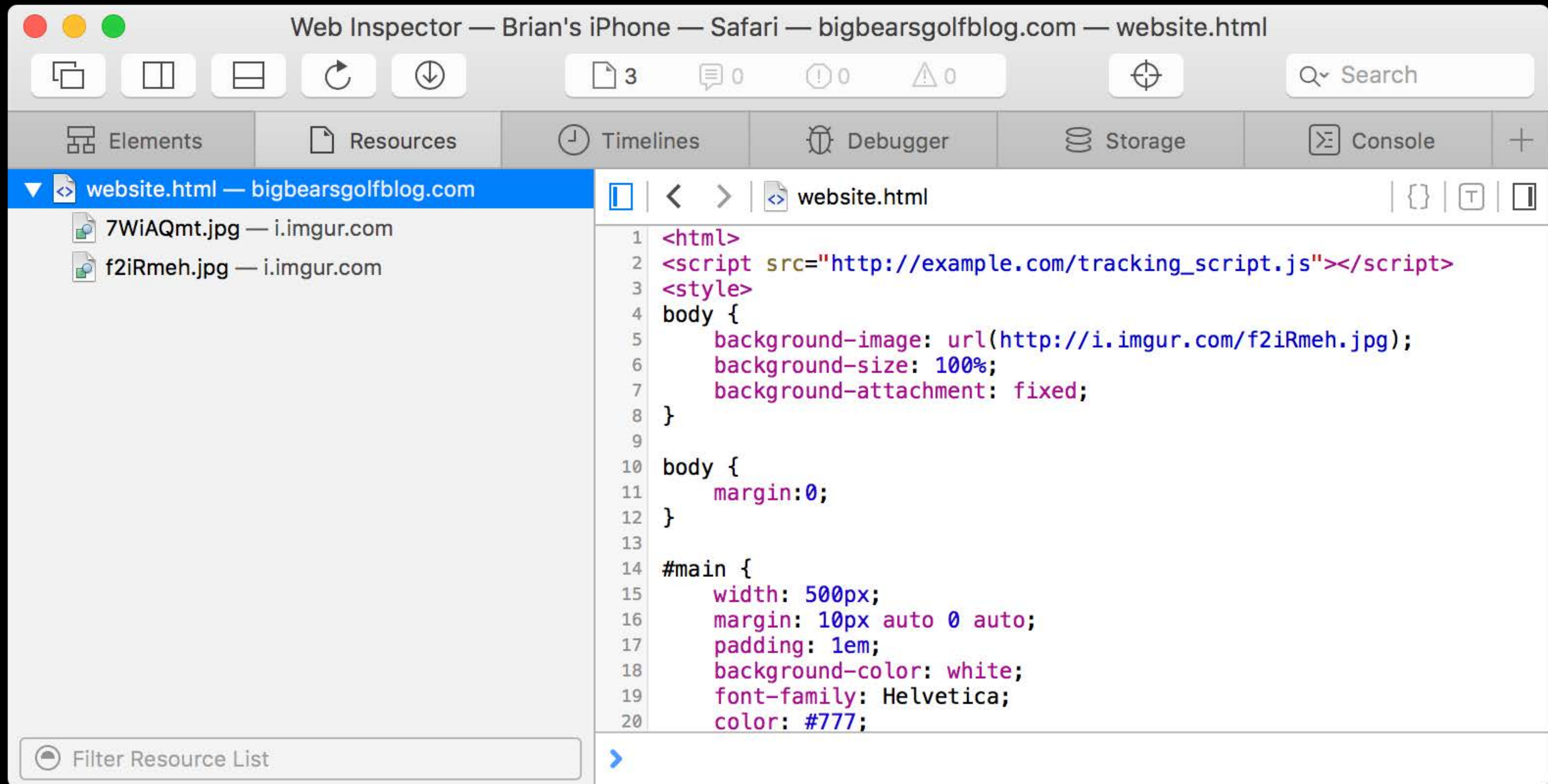
Block a load





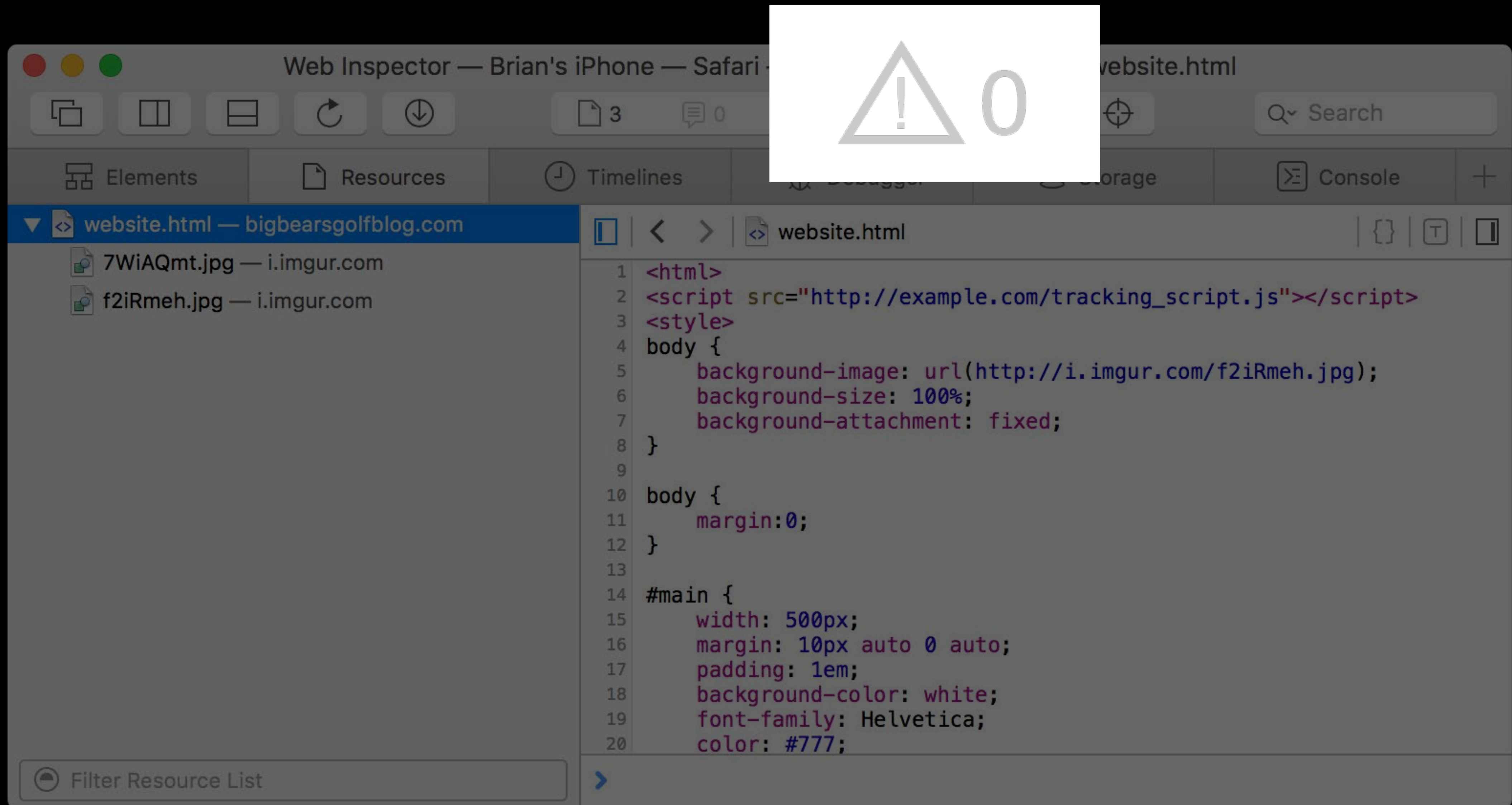
# What Content Blockers Can Do

Block a load



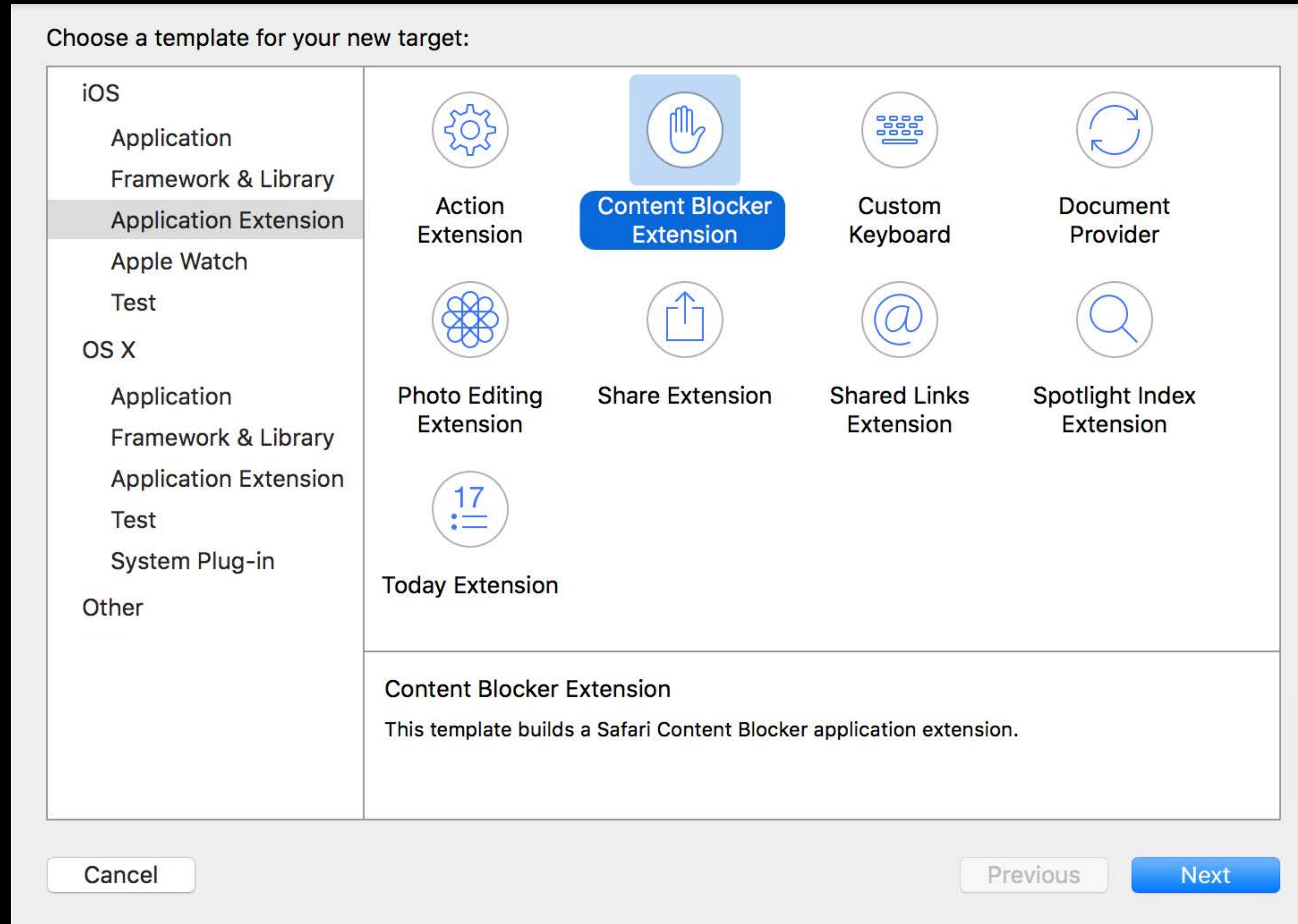
# What Content Blockers Can Do

Block a load





# Creating Content Blockers on iOS



# Creating Content Blockers on iOS

```
func beginRequestWithExtensionContext(context: NSExtensionContext) {  
    let item = NSExtensionItem()  
  
    let blockerURL = NSBundle.mainBundle().URLForResource("blockerList",  
        withExtension: "json")  
    item.attachments = [ NSItemProvider(contentsOfURL: blockerURL)! ]  
  
    context.completeRequestReturningItems([ item ], completionHandler: nil)  
}
```

# Creating Content Blockers on iOS

```
func beginRequestWithExtensionContext(context: NSExtensionContext) {  
    let item = NSExtensionItem()  
  
    let blockerURL = NSBundle.mainBundle().URLForResource("blockerList",  
        withExtension: "json")  
    item.attachments = [ NSItemProvider(contentsOfURL: blockerURL)! ]  
  
    context.completeRequestReturningItems([ item ], completionHandler: nil)  
}
```

# Updating Content Blockers

Management settings in your app

API in the SafariServices Framework

# Updating Content Blockers

Management settings in your app  
API in the SafariServices Framework

# Updating Content Blockers

Management settings in your app

API in the SafariServices Framework

```
SFContentBlockerManager.reloadContentBlockerWithIdentifier(  
    identifier: String,  
    completionHandler: ((NSError?) -> Void)?  
)
```

*Demo*

Content Blockers on iOS

Alex Christensen

Safari and WebKit Engineer



# Content Blocking on OS X

NEW

Brought the same model to the Mac

# Content Blocking on OS X

NEW

Brought the same model to the Mac

Faster and more memory efficient

# Content Blocking on OS X

NEW


Brought the same model to the Mac


Faster and more memory efficient

**canLoad** has been deprecated

# Content Blocking on OS X

## Extension Builder

Content Blocker

**Content Blocker**

Content Blocker.safariextension

Safari Developer: (A1B2C3D4E5) admin@example.com

Install

Build Package...

**Versions**

Display Version: 1.0

Bundle Version: 1

**Website Access**

Access Level: None

**Content Blocker**

Content Blocker File: None

**Global Page**

Global Page File: None

**Storage**


Database Quota: None

**Chrome**

# Content Blocking on OS X

## Extension Builder

Content Blocker



**Content Blocker**

Content Blocker.safariextension

Safari Developer: (A1B2C3D4E5) admin@example.com

Install

Build Package...

**Versions**

Display Version:

1.0

Bundle Version:

1

**Website Access**

Access Level:

None

**Content Blocker**

Content Blocker File:

None

**Global Page**

Global Page File:

None

**Storage**

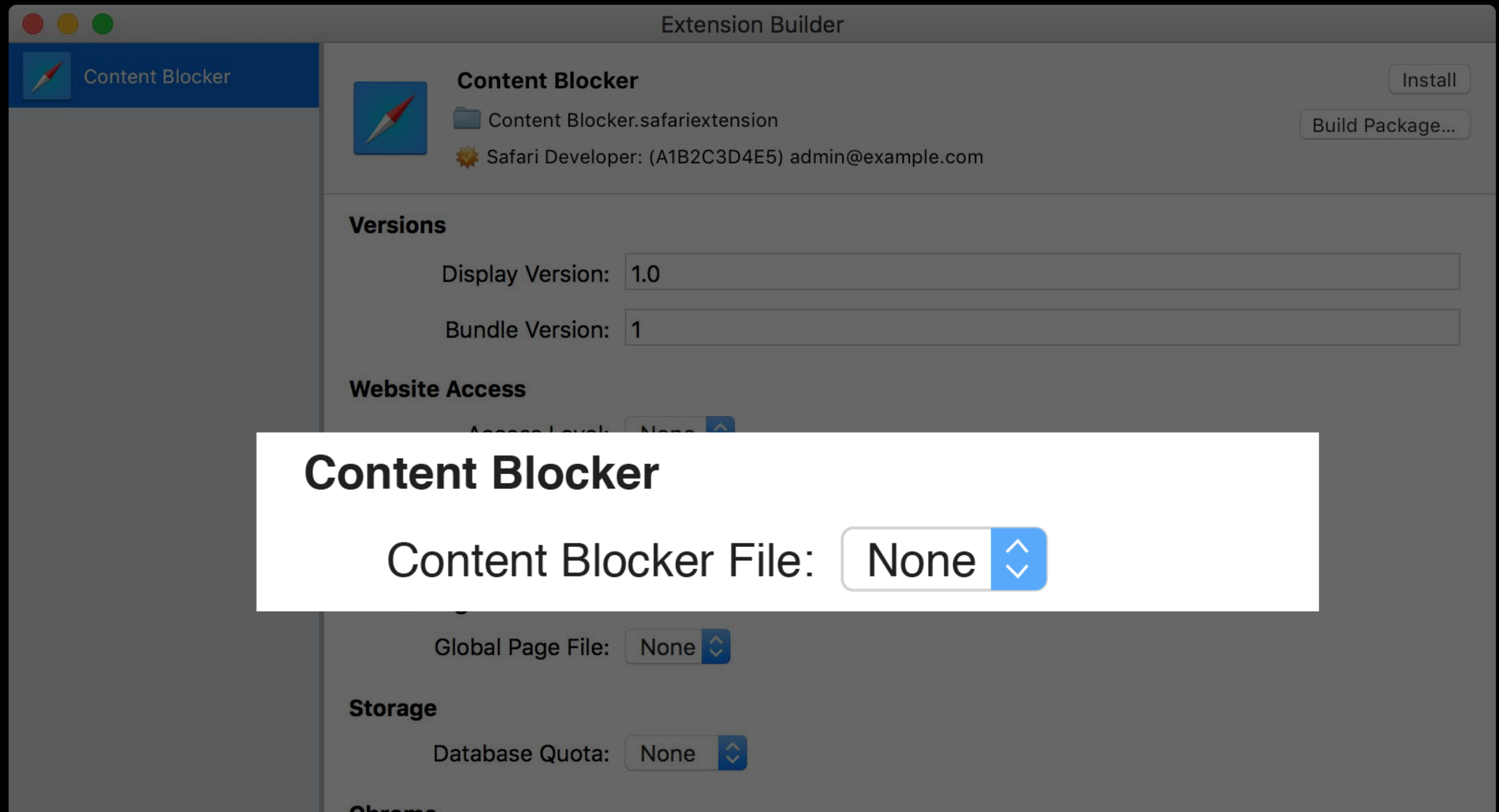
Database Quota:

None

**Chrome**

# Content Blocking on OS X

## Extension Builder



# Content Blocking on OS X

JavaScript API



# Content Blocking on OS X

JavaScript API

```
safari.extension.setContentBlocker(contentBlocker)
```

# Content Blocking on OS X

## JavaScript API

```
safari.extension.setContentBlocker(contentBlocker)
```

- **contentBlocker** can either be an object or a JSON string

# What We Are Going to Talk About

NEW

Content Blockers

Changes to the Extensions Gallery

Shared Links App Extensions

# What We Are Going to Talk About

NEW

Content Blockers

Changes to the Extensions Gallery

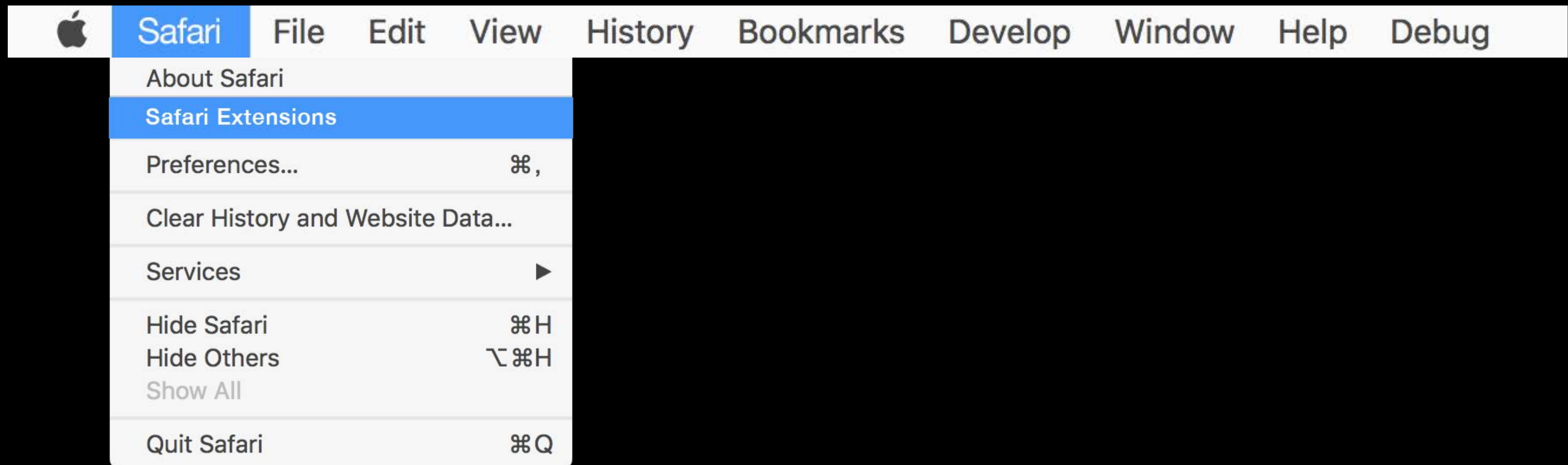
Shared Links App Extensions

# Changes to the Extensions Gallery



# Safari Extension Updates

## Extensions Gallery



# Safari Extension Updates

## Extensions Gallery



# Safari Extension Updates

## Extensions Gallery



### ColorFinder

Productivity

[Jessie Berlin >](#)

Shows all the background, text, and border colors used on the page.

[Install Now](#) 

# Safari Extension Updates

## Extensions Gallery

### New Extension Guidelines

- Extensions will now be signed and hosted by Apple
- Please resubmit to the gallery

# Safari Extension Updates

## Extensions Gallery

### New Extension Guidelines

- Extensions will now be signed and hosted by Apple
- Please resubmit to the gallery

### New flag in update manifest

```
<key>Update From Gallery</key>  
<true/>
```

# What We Are Going to Talk About

NEW

Content Blockers

Changes to the Extensions Gallery

Shared Links App Extensions



# What We Are Going to Talk About

NEW

Content Blockers

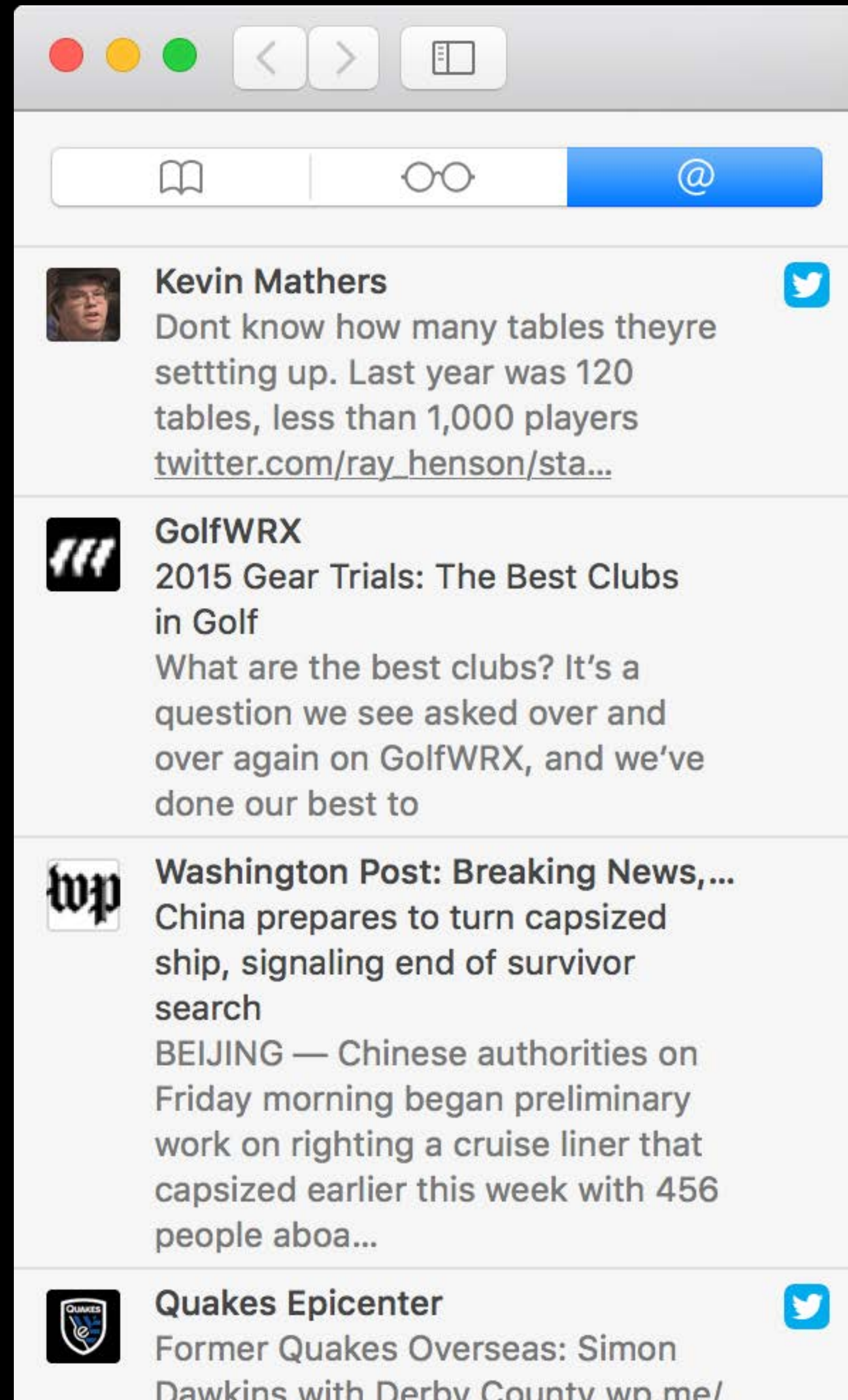
Changes to the Extensions Gallery

Shared Links App Extensions

# Shared Links App Extensions

iOS and OS X

# Shared Links



# Shared Links App Extensions

NEW

For apps to put their content in Safari's Shared Links

Same API on iOS and OS X

# Shared Links App Extensions

NEW

For apps to put their content in Safari's Shared Links

Same API on iOS and OS X



# Creating a Shared Links App Extension

Choose a template for your new target:

iOS

Application

Framework & Library

Application Extension

Apple Watch

Test

OS X

Application

Framework & Library

Application Extension

Test

System Plug-in

Other

Action Extension

Content Blocker Extension

Custom Keyboard

Document Provider

Photo Editing Extension

Share Extension

Shared Links Extension

Spotlight Index Extension

Today Extension

Shared Links Extension

This template builds a Safari Shared Links application extension.

Cancel

Previous

Next



# Creating a Shared Links App Extension

```
let extensionItem = NSExtensionItem()
```

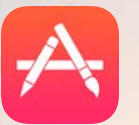
# Creating a Shared Links App Extension

```
let extensionItem = NSExtensionItem()
```



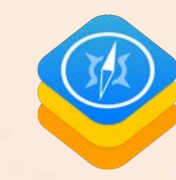
**WebKit News**

WebKit is awesome  
Here are some reasons  
why WebKit is awesome.



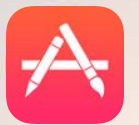
# Creating a Shared Links App Extension

```
extensionItem.userInfo = [  
    "uniqueIdentifier": "A1B2-C3D4"
```



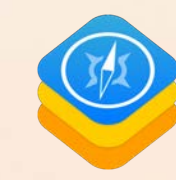
## WebKit News

WebKit is awesome  
Here are some reasons  
why WebKit is awesome.



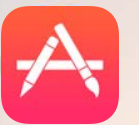
# Creating a Shared Links App Extension

```
extensionItem.userInfo = [  
    "uniqueIdentifier": "A1B2-C3D4",  
    "urlString": "http://webkit.org"
```



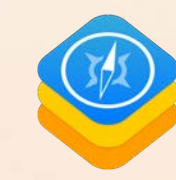
## WebKit News

WebKit is awesome  
Here are some reasons  
why WebKit is awesome.



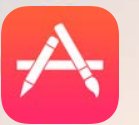
# Creating a Shared Links App Extension

```
extensionItem.userInfo = [  
    "uniqueIdentifier": "A1B2-C3D4",  
    "urlString": "http://webkit.org",  
    "date": NSDate(  
        timeIntervalSinceReferenceDate:  
        455474512)
```



## WebKit News

WebKit is awesome  
Here are some reasons  
why WebKit is awesome.



# Creating a Shared Links App Extension

```
extensionItem.userInfo = [  
    "uniqueIdentifier": "A1B2-C3D4",  
    "urlString": "http://webkit.org",  
    "date": NSDate(  
        timeIntervalSinceReferenceDate:  
        455474512),  
    "displayName": "WebKit News"  
]
```

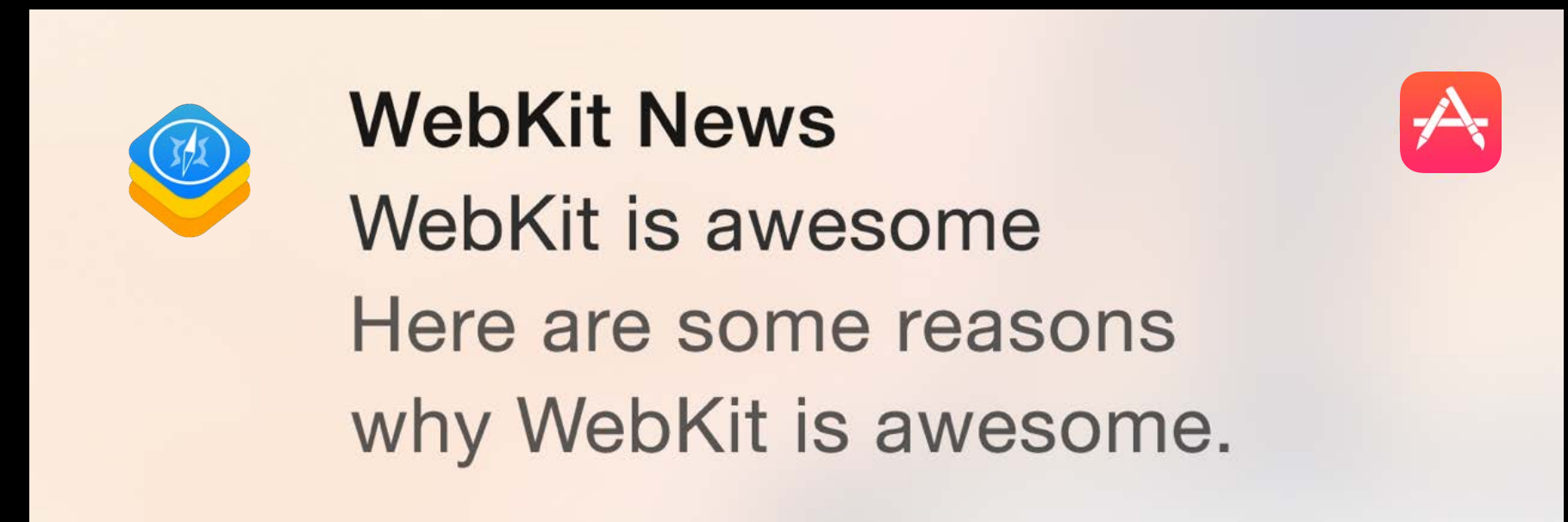




# Creating a Shared Links App Extension

```
extensionItem.userInfo = [  
    "uniqueIdentifier": "A1B2-C3D4",  
    "urlString": "http://webkit.org",  
    "date": NSDate(  
        timeIntervalSinceReferenceDate:  
        455474512),  
    "displayName": "WebKit News"  
]
```

**uniqueIdentifier** must be consistent  
across instantiations of the extension



# Creating a Shared Links App Extension

```
extensionItem.userInfo = [  
    "uniqueIdentifier": "A1B2-C3D4",  
    "urlString": "http://webkit.org",  
    "date": NSDate(  
        timeIntervalSinceReferenceDate:  
        455474512),  
    "displayName": "WebKit News"  
]
```

`uniqueIdentifier` must be consistent  
across instantiations of the extension

If `displayName` is not set, Safari falls back to  
display name of the App Extension



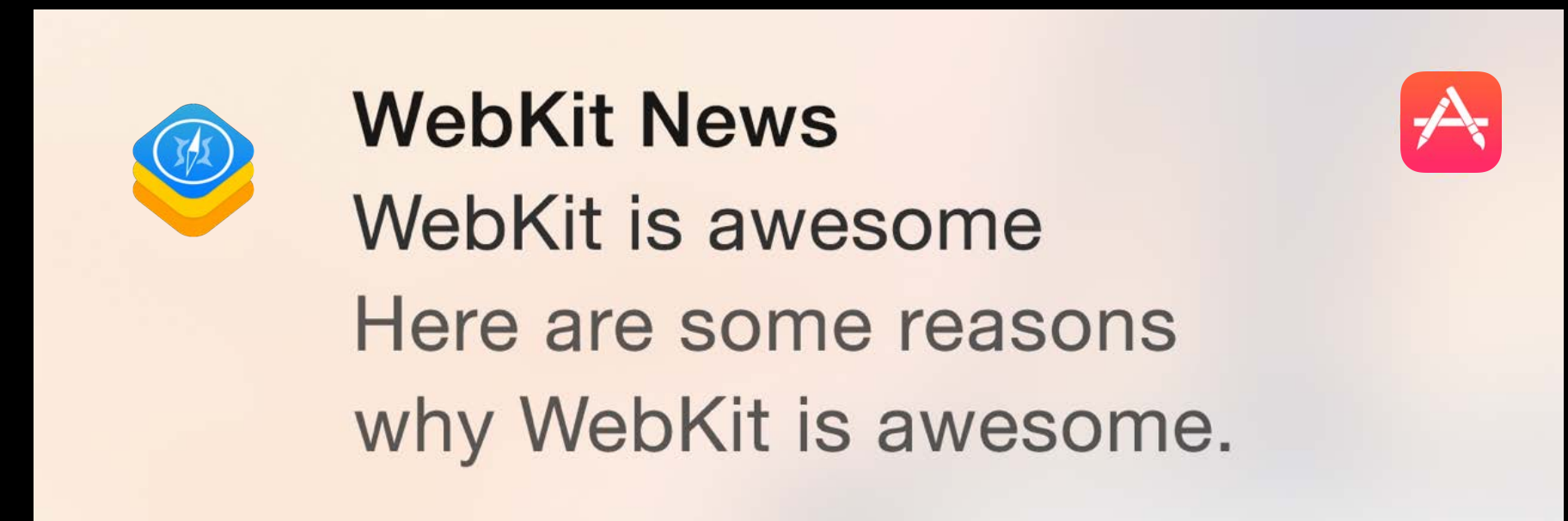
# Creating a Shared Links App Extension

```
extensionItem.attributedTitle =  
    NSAttributedString(string: "WebKit  
is awesome")
```



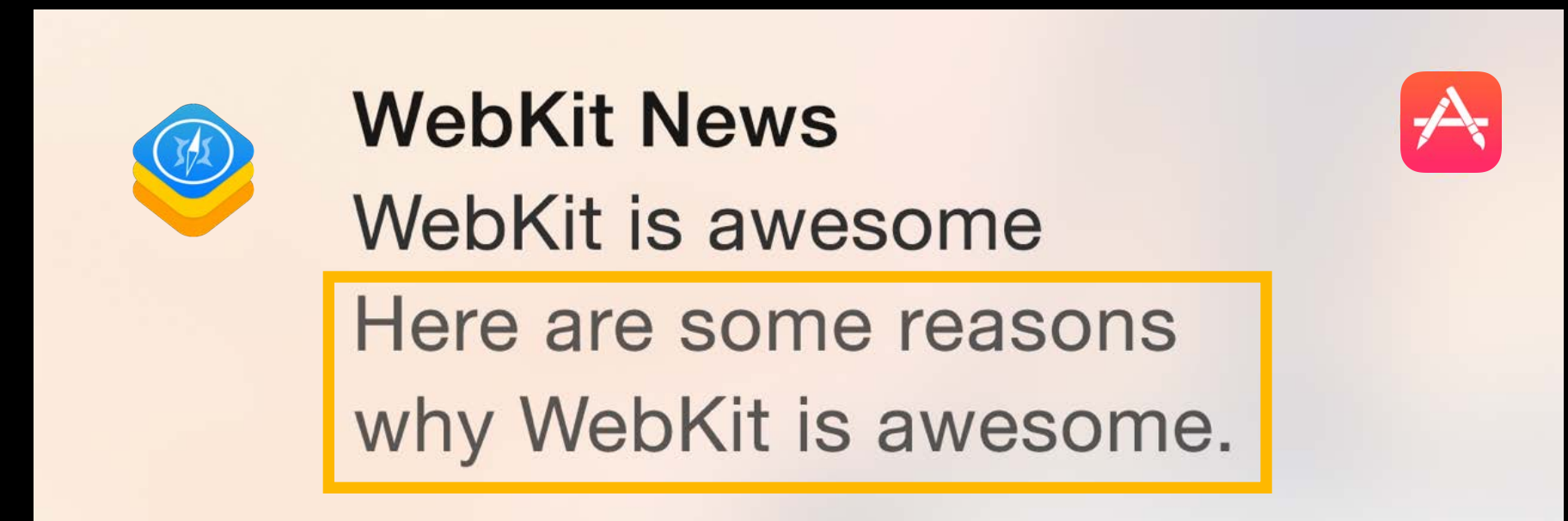
# Creating a Shared Links App Extension

```
extensionItem.attributedTitle =  
    NSAttributedString(string: "WebKit  
    is awesome")  
  
extensionItem.attributedContentText =  
    NSAttributedString(string: "Here are  
    some reasons why WebKit is  
    awesome.")
```



# Creating a Shared Links App Extension

```
extensionItem.attributedTitle =  
    NSAttributedString(string: "WebKit  
    is awesome")  
  
extensionItem.attributedContentText =  
    NSAttributedString(string: "Here are  
    some reasons why WebKit is  
    awesome.")
```

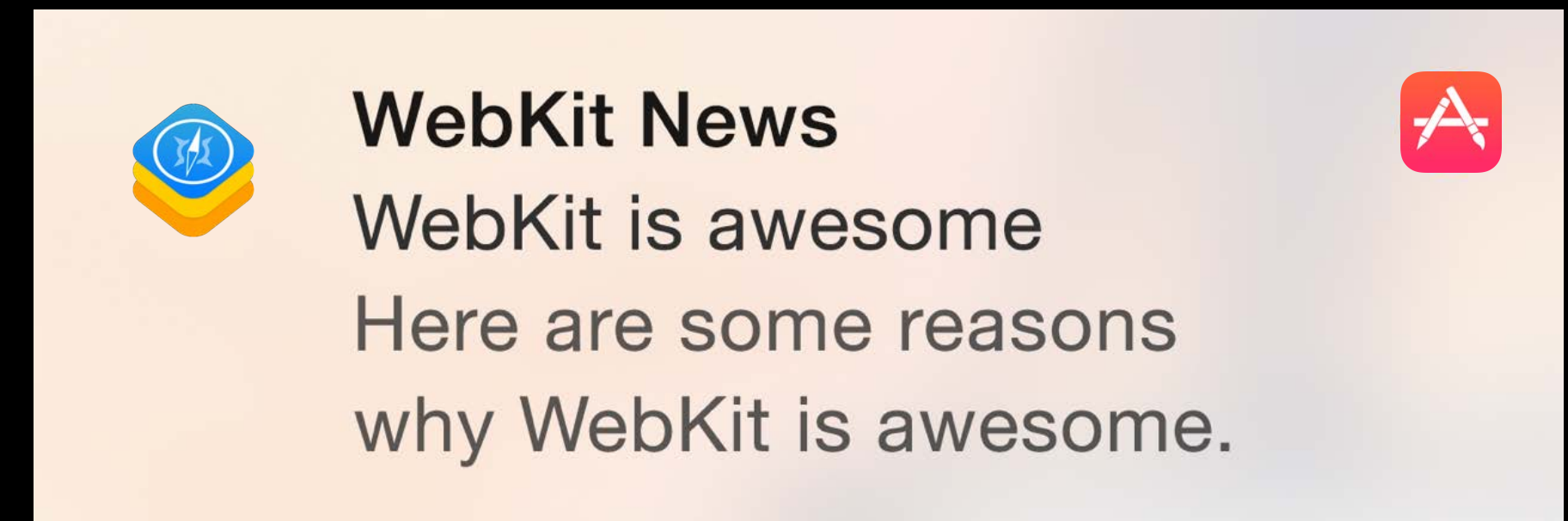




# Creating a Shared Links App Extension

```
extensionItem.attributedTitle =  
    NSAttributedString(string: "WebKit  
    is awesome")  
  
extensionItem.attributedContentText =  
    NSAttributedString(string: "Here are  
    some reasons why WebKit is  
    awesome.")
```

Title and content text will be automatically  
ellipsized if they are too long



# Creating a Shared Links App Extension



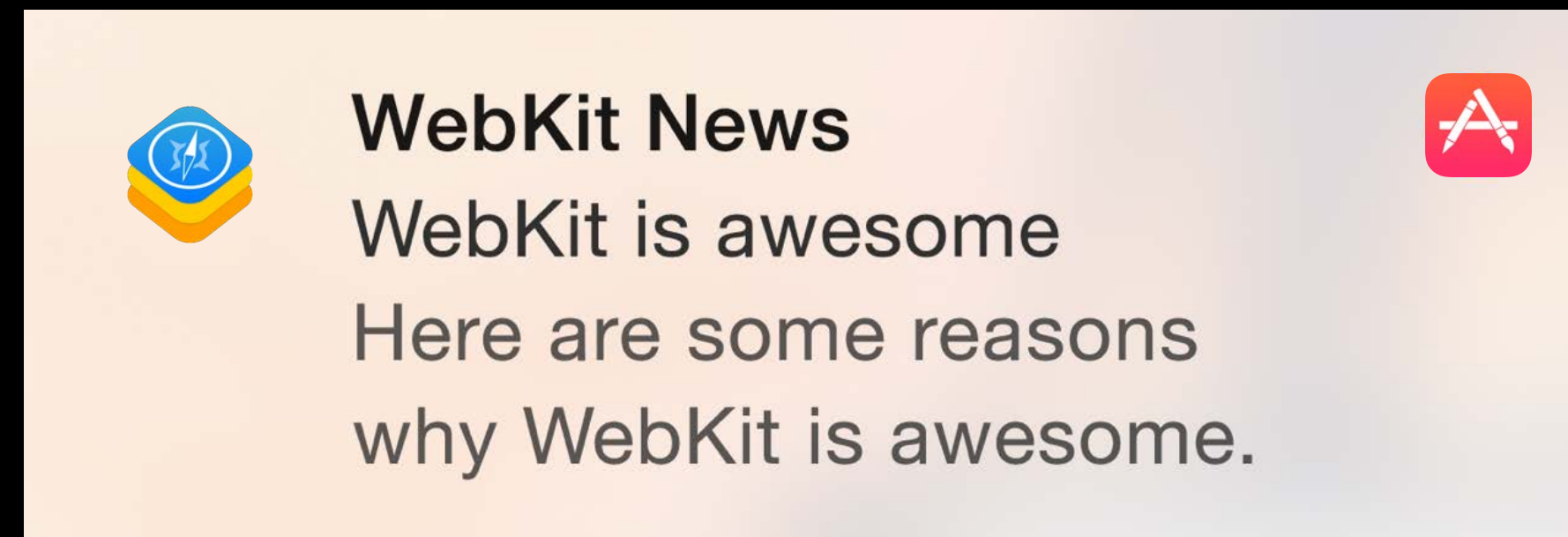
```
let bundle = NSBundle.mainBundle()
let iconURL = bundle.URLForResource("icon", withExtension: "png")
extensionItem.attachments = [ NSItemProvider(contentsOfURL: iconURL )! ]
```

# Creating a Shared Links App Extension



```
let bundle = NSBundle.mainBundle()  
let iconURL = bundle.URLForResource("icon", withExtension: "png")  
extensionItem.attachments = [ NSItemProvider(contentsOfURL: iconURL )! ]
```

# Creating a Shared Links App Extension



```
let bundle = NSBundle.mainBundle()
let iconURL = bundle.URLForResource("icon", withExtension: "png")
extensionItem.attachments = [ NSItemProvider(contentsOfURL: iconURL )! ]
```

Icon in top right corner is your app's icon

# Creating a Shared Links App Extension



```
let bundle = NSBundle.mainBundle()  
let iconURL = bundle.URLForResource("icon", withExtension: "png")  
extensionItem.attachments = [ NSItemProvider(contentsOfURL: iconURL )! ]
```

Icon in top right corner is your app's icon



# Creating a Shared Links App Extension



```
let bundle = NSBundle.mainBundle()  
let iconURL = bundle.URLForResource("icon", withExtension: "png")  
extensionItem.attachments = [ NSItemProvider(contentsOfURL: iconURL )! ]
```

Icon in top right corner is your app's icon

All properties must be set after **userInfo**



*Demo*

Creating a Shared Links App Extension

Alex Christensen

Safari and WebKit Engineer

# Summary

Write content blockers

Submit your extensions to the Safari Extensions Gallery

Put your content in Shared Links

# Summary

Write content blockers

Submit your extensions to the Safari Extensions Gallery

Put your content in Shared Links

# Summary

Write content blockers

Submit your extensions to the Safari Extensions Gallery

Put your content in Shared Links

# More Information

## Technical Support

Apple Developer Forums

<http://developer.apple.com/forums>

<http://webkit.org/blog/>

## General Inquiries

Jon Davis, Web Technologies Evangelist

[web-evangelist@apple.com](mailto:web-evangelist@apple.com)



# Related Sessions

Introducing Safari View Controller	Nob Hill	Tuesday 1:30PM
Using Safari to Deliver and Debug a Responsive Web Design	Mission	Wednesday 9:00AM
What's New in Network Extension and VPN	Nob Hill	Friday 9:00AM

# Related Labs

---

Safari and WebKit Lab

Graphics, Games,  
and Media Lab A

Friday 12:00PM

---

