

FLEXBOX & GRID

— BASIS, GROW & SHRINK —



IN A ROCKET

Learn front-end development at rocket speed

← **FLEX-GROW** →



← **FLEX-BASIS** →

→ **FLEX-SHRINK** ←

FLEX-BASIS



FLEX-GROW



FLEX-SHRINK



FLEX-BASIS



FLEX-GROW

FLEX-SHRINK

“**Flex basis:** the initial main size of the flex item, before free space is distributed according to the flex factors.”



SOURCE: [CSS Flexible Box Layout Module Level 1](#) by W3C.

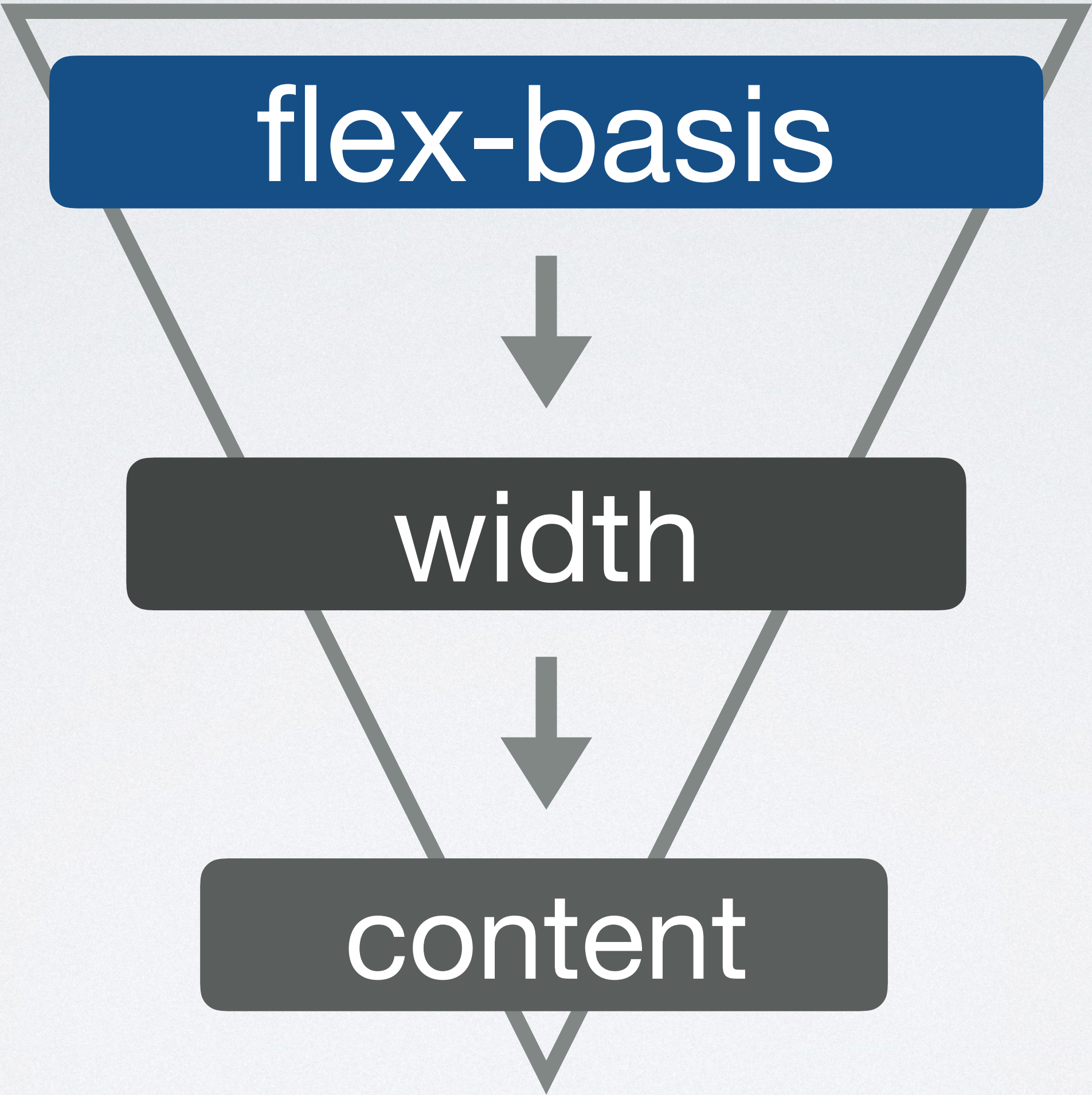
flex-basis

auto (default)

content

width

FLEX-BASIS: PRIORITY IN FLEXBOX



FLEX-BASIS

HTML

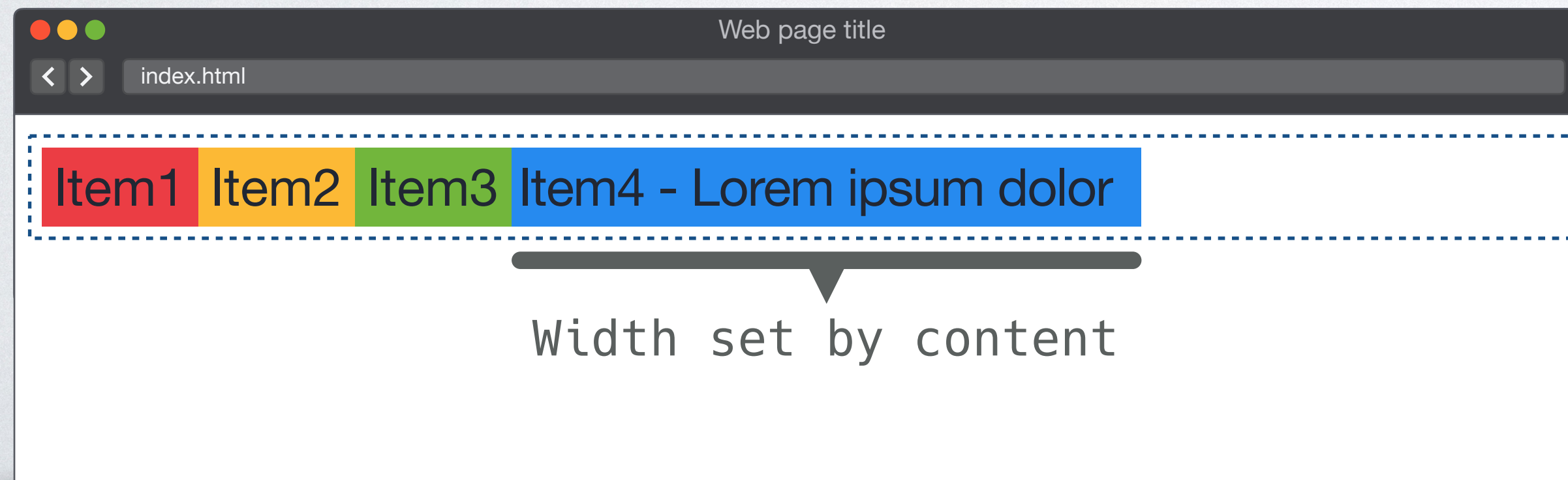
```
<body>
<div class="container">
  <div>Item1</div>
  <div>Item2</div>
  <div>Item3</div>
  <div class="i4">Item4</div>
</div>
</body>
```

CSS

```
.container {display: flex;}

.i4 {
  flex-basis: auto; /* Or not defined */
}
```

Browser



FLEX-BASIS

HTML

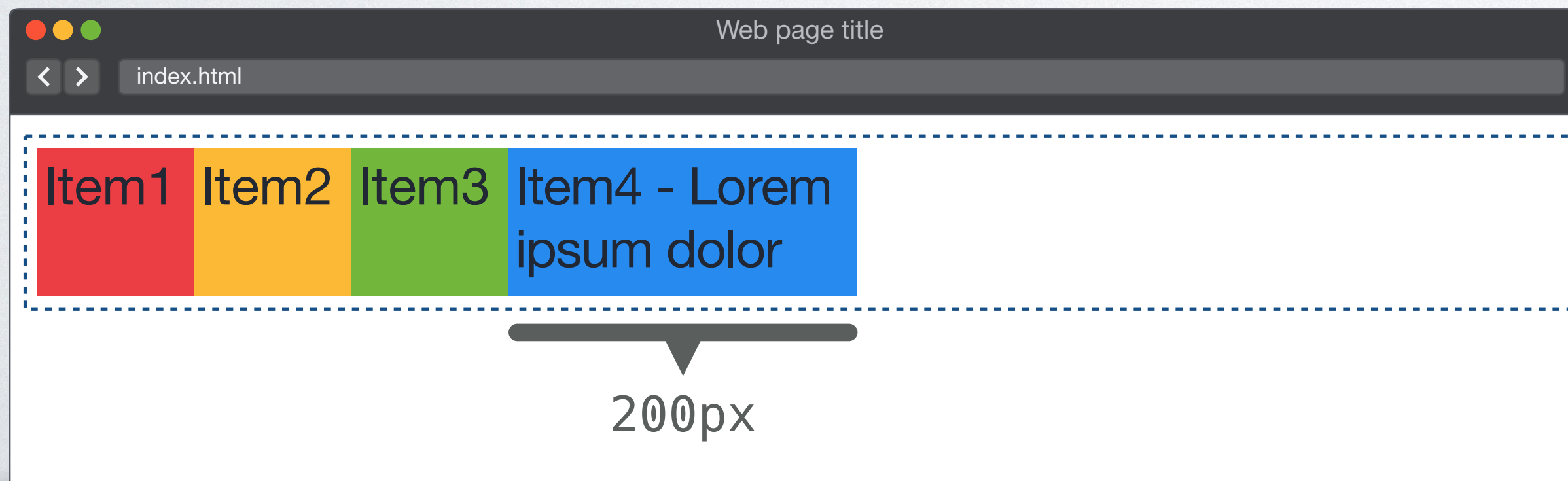
```
<body>
<div class="container">
  <div>Item1</div>
  <div>Item2</div>
  <div>Item3</div>
  <div class="i4">Item4</div>
</div>
</body>
```

CSS

```
.container {display: flex;}

.i4 {
  flex-basis: auto; /* Or no defined */
  width: 200px; ← Prevails
}
```

Browser



FLEX-BASIS

HTML

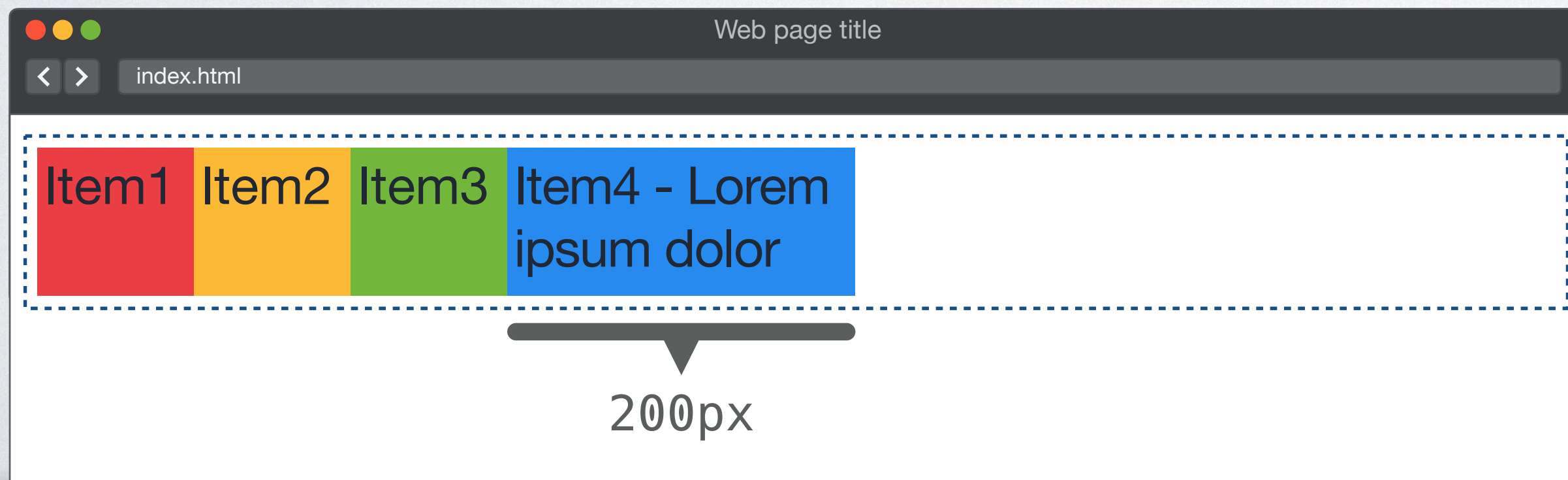
```
<body>
<div class="container">
  <div>Item1</div>
  <div>Item2</div>
  <div>Item3</div>
  <div class="i4">Item4</div>
</div>
</body>
```

CSS

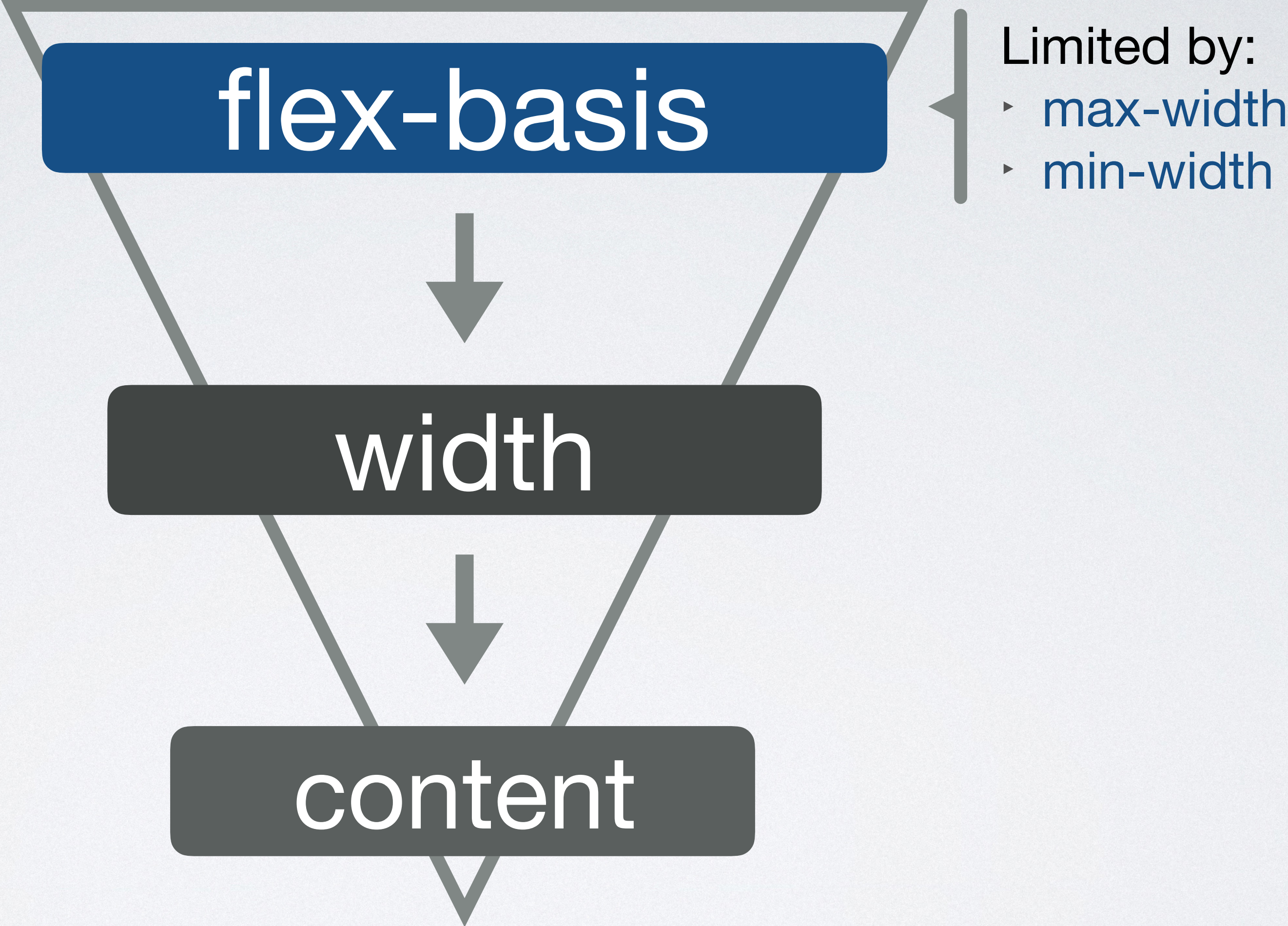
```
.container {display: flex;}

.i4 {
  flex-basis: 200px; ← Prevails
  width: 50px;
}
```

Browser



FLEX-BASIS: PRIORITY IN FLEXBOX



FLEX-BASIS

HTML

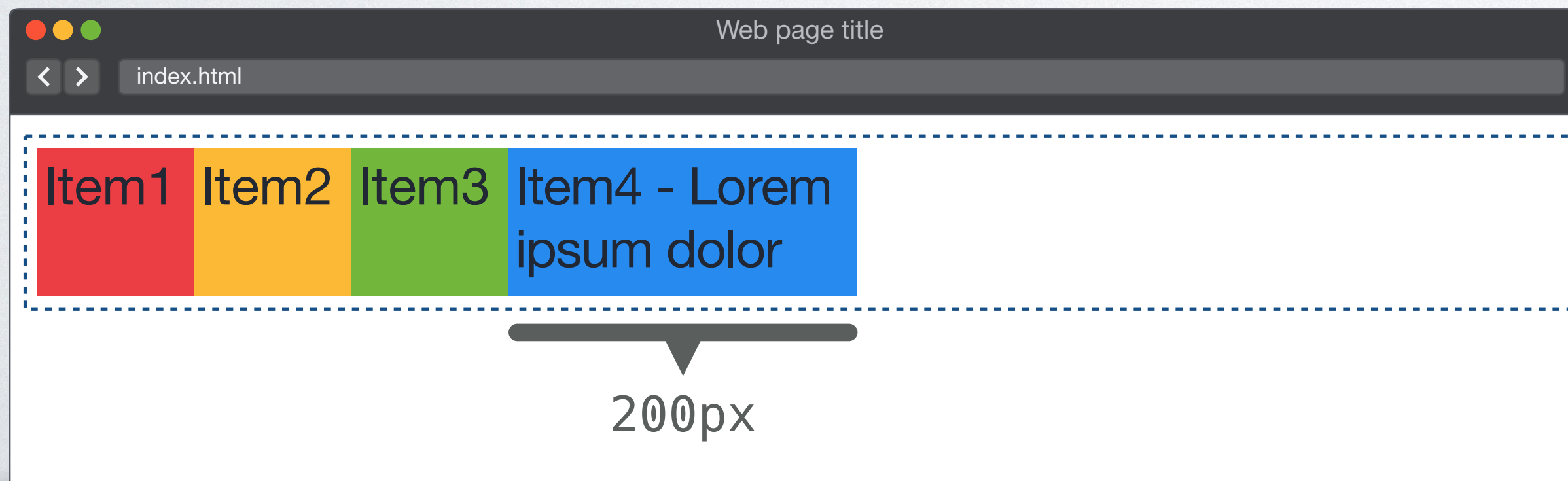
```
<body>
<div class="container">
  <div>Item1</div>
  <div>Item2</div>
  <div>Item3</div>
  <div class="i4">Item4</div>
</div>
</body>
```

CSS

```
.container {display: flex;}

.i4 {
  flex-basis: 5000px;
  max-width: 200px; ← Sets limit
}
```

Browser



FLEX-BASIS

HTML

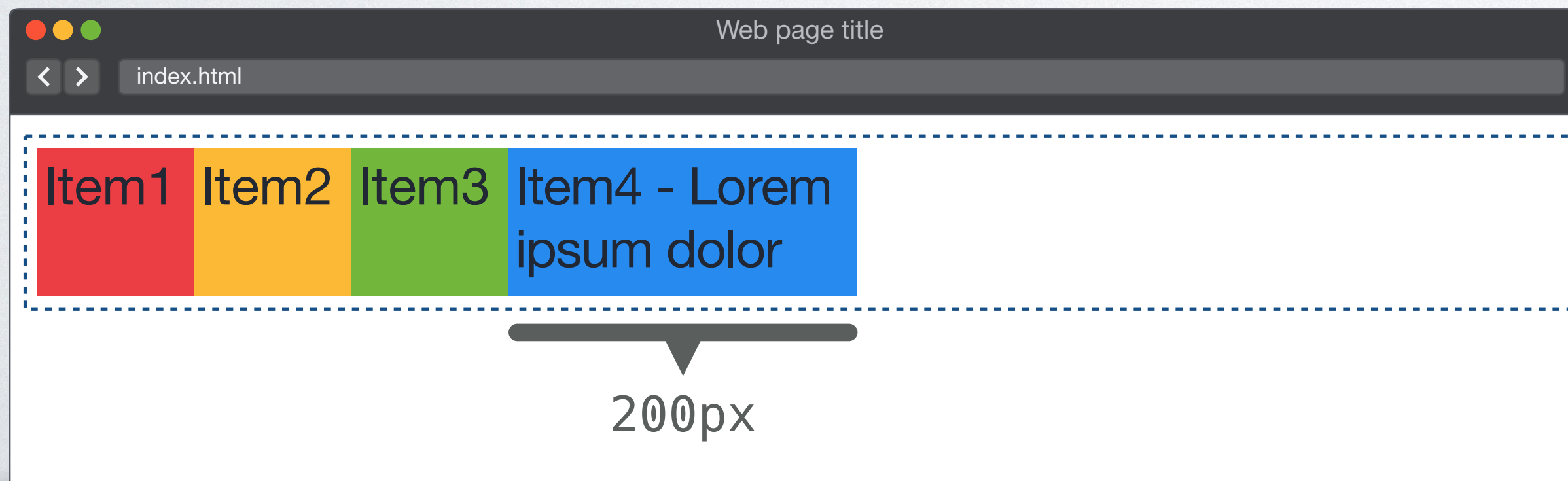
```
<body>
<div class="container">
  <div>Item1</div>
  <div>Item2</div>
  <div>Item3</div>
  <div class="i4">Item4</div>
</div>
</body>
```

CSS

```
.container {display: flex;}

.i4 {
  flex-basis: 50px;
  min-width: 200px; ← Sets limit
}
```

Browser



FLEX-BASIS

FLEX-GROW



FLEX-SHRINK

“**Flex basis:** the initial main size of the flex item,
before free space is distributed according to the flex factors.”



SOURCE: [CSS Flexible Box Layout Module Level 1](#) by W3C.

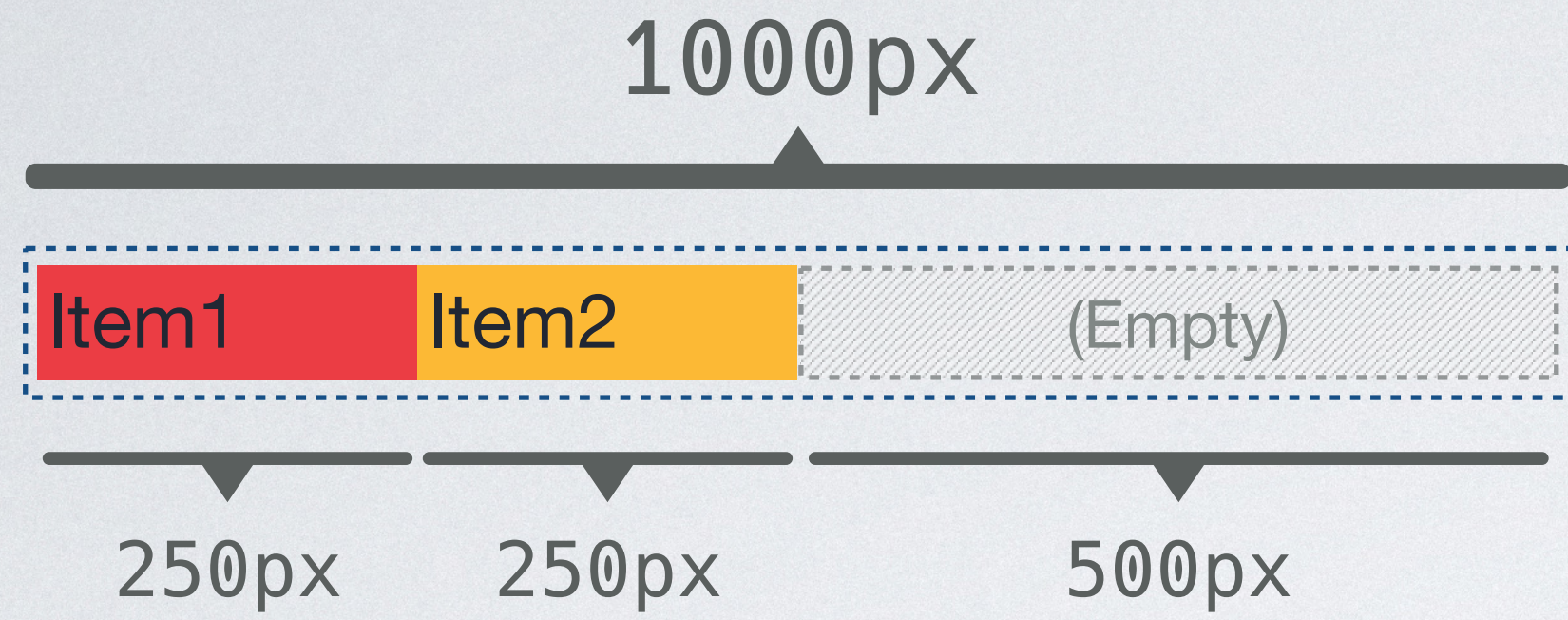
flex-grow

number (default: 0)

Not negative

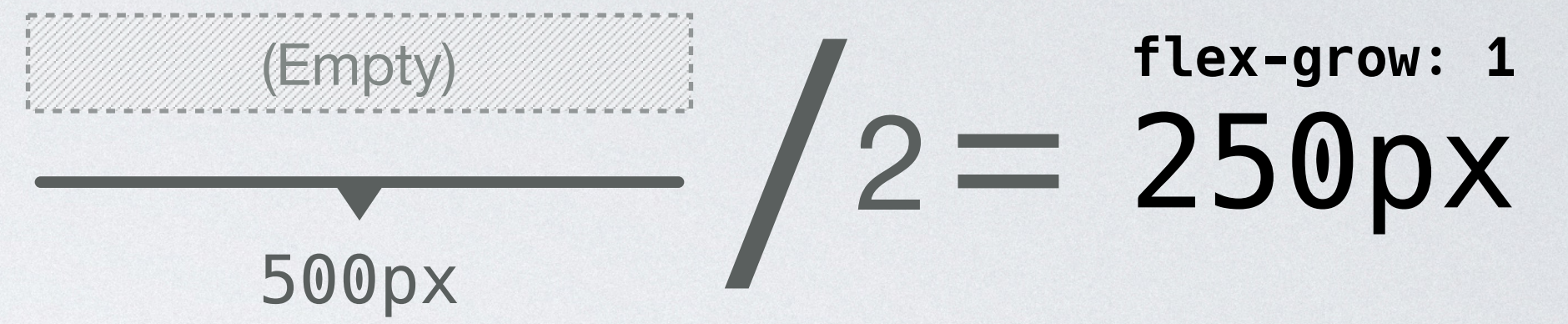
1

```
.i1 {flex-basis: 250px; flex-grow: 0 /* Default*/}
.i2 {flex-basis: 250px; flex-grow: 0 /* Default*/}
```



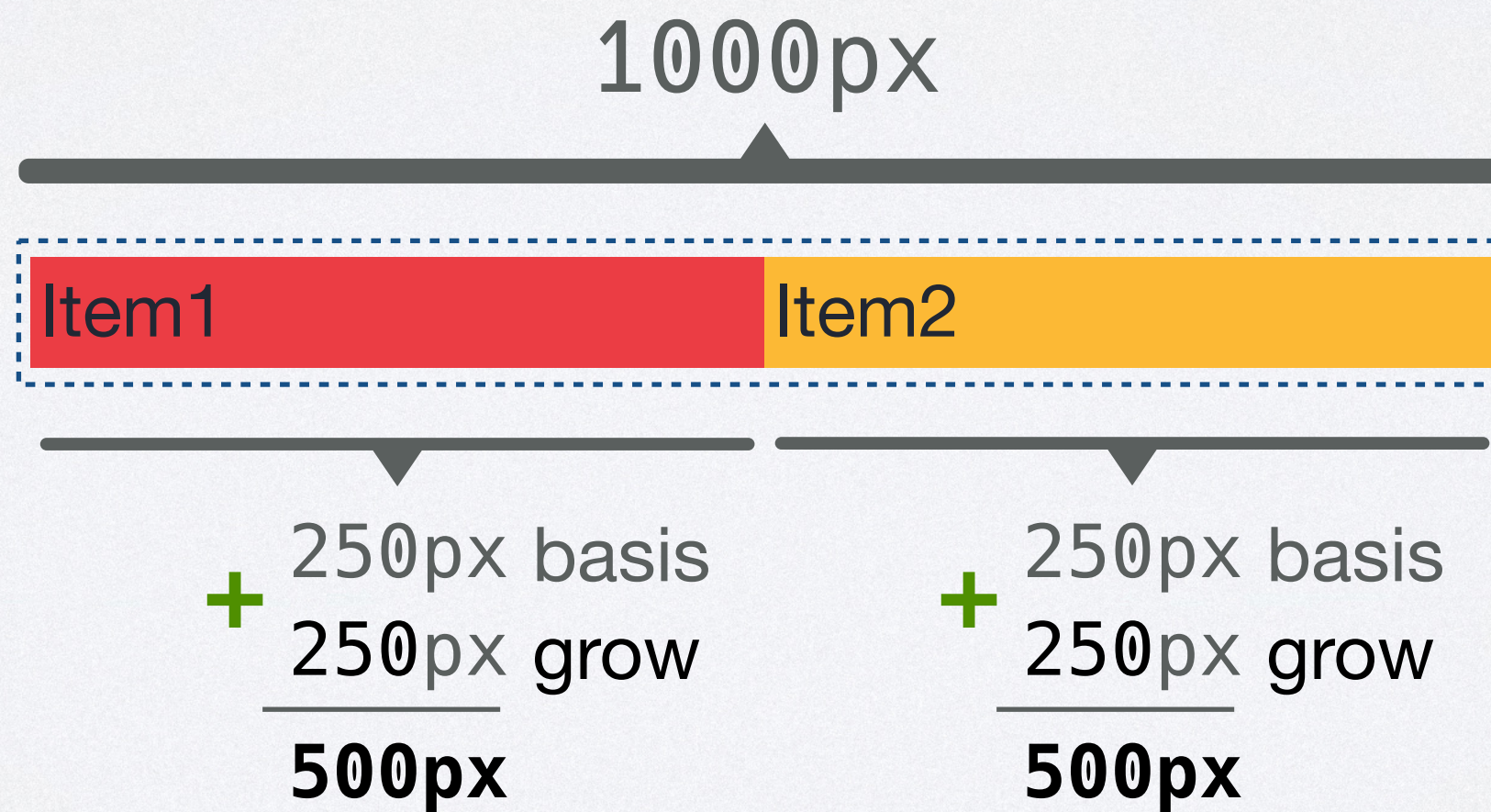
2

```
.i1 {flex-basis: 250px; flex-grow: 1}
.i2 {flex-basis: 250px; flex-grow: 1} ▶ 2
```



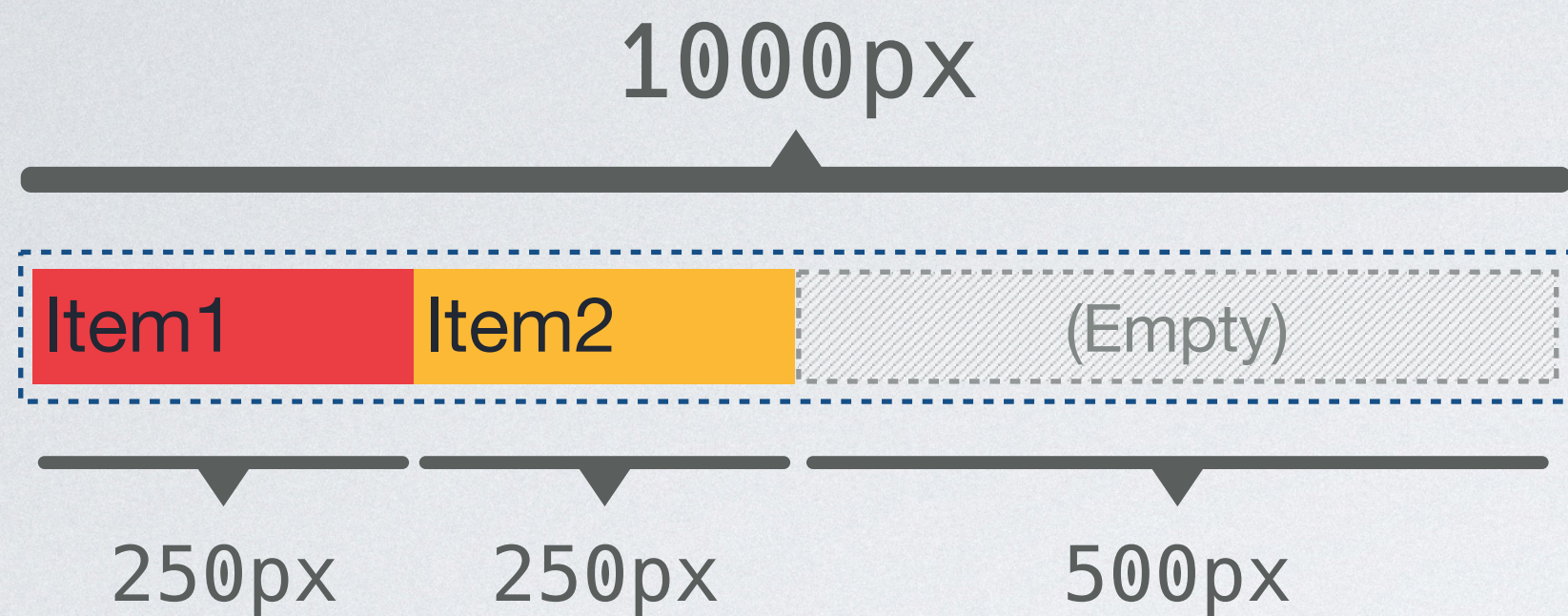
3

```
.i1 {flex-basis: 250px; flex-grow: 1}
.i2 {flex-basis: 250px; flex-grow: 1}
```



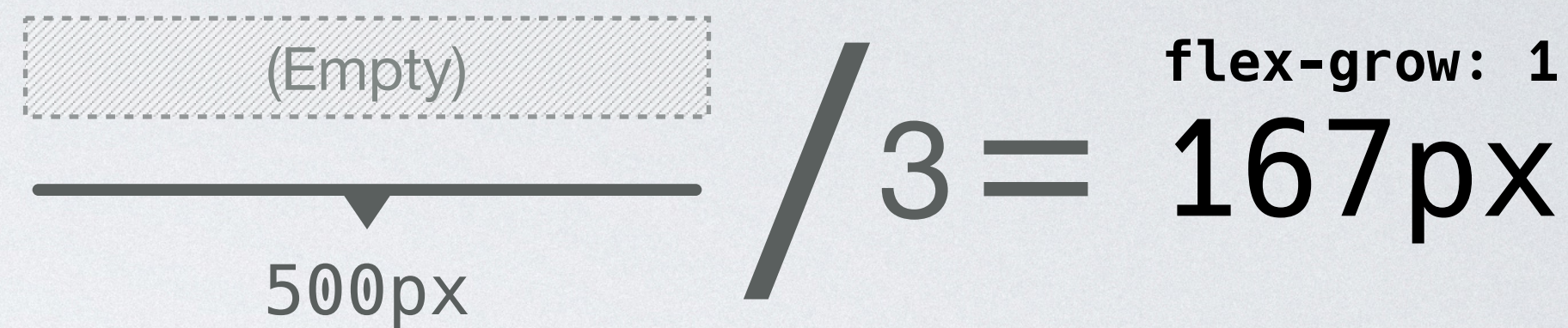
1

```
.i1 {flex-basis: 250px; flex-grow: 0 /* Default*/}
.i2 {flex-basis: 250px; flex-grow: 0 /* Default*/}
```



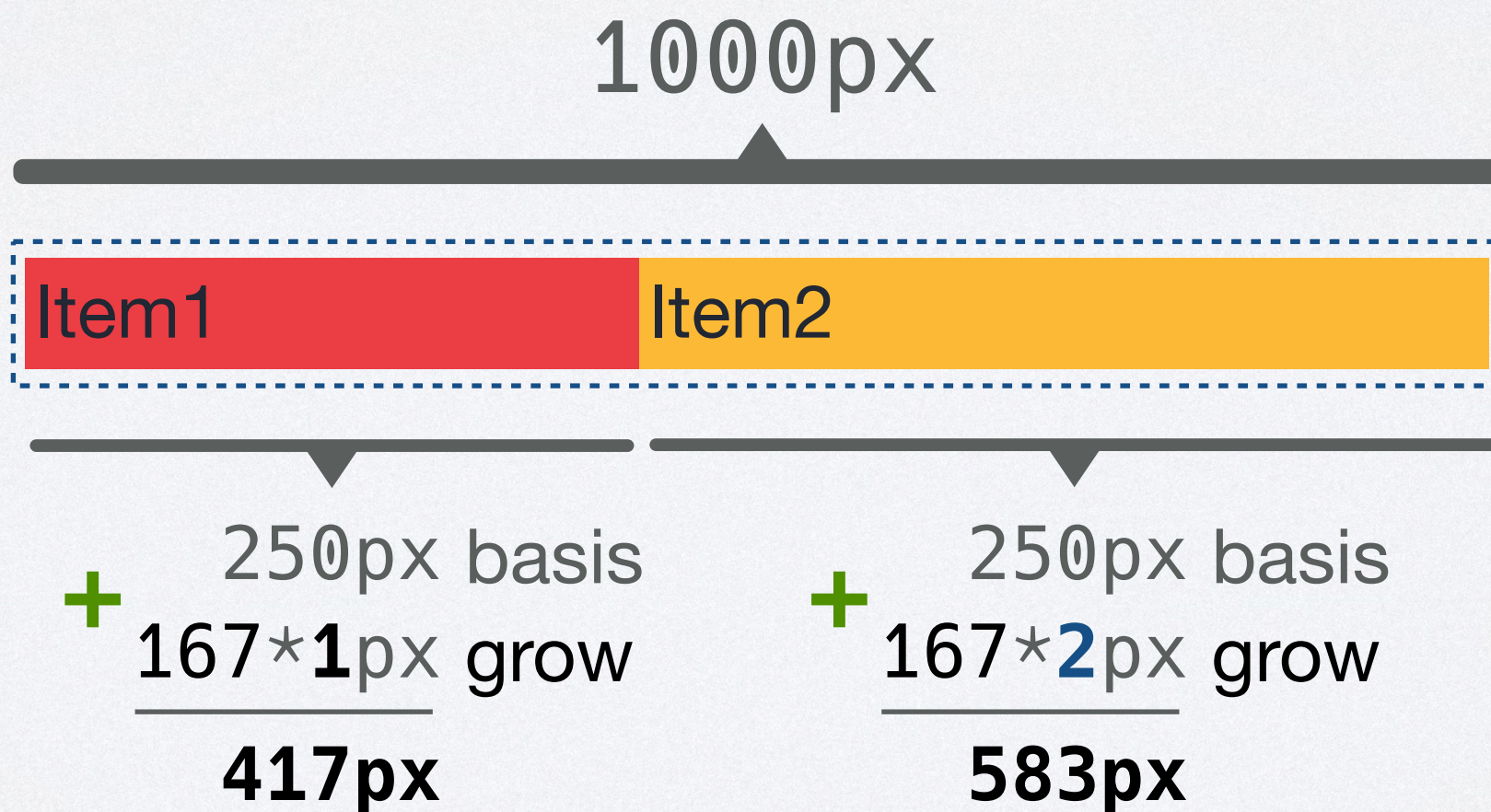
2

```
.i1 {flex-basis: 250px; flex-grow: 1}
.i2 {flex-basis: 250px; flex-grow: 2} ▶ 3
```



3

```
.i1 {flex-basis: 250px; flex-grow: 1}
.i2 {flex-basis: 250px; flex-grow: 2}
```



HTML

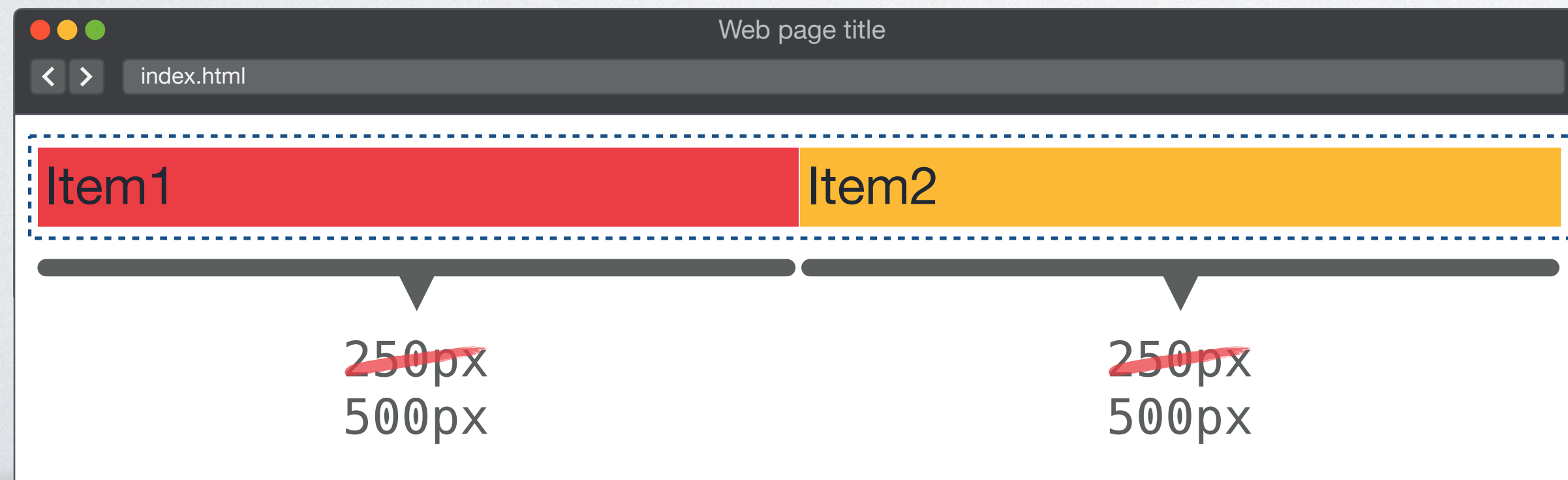
```
<body>
<div class="container">
  <div class="i1">Item1</div>
  <div class="i2">Item2</div>
</div>
</body>
```

CSS

```
.container {
  display: flex;
  width: 1000px;
}

.i1, .i2 {
  flex-basis: 250px;
  flex-grow: 1;
}
```

Browser



1000px

With flex-grow the items fill the available space.
flex-grow: 1

FLEX-BASIS

FLEX-GROW

FLEX-SHRINK



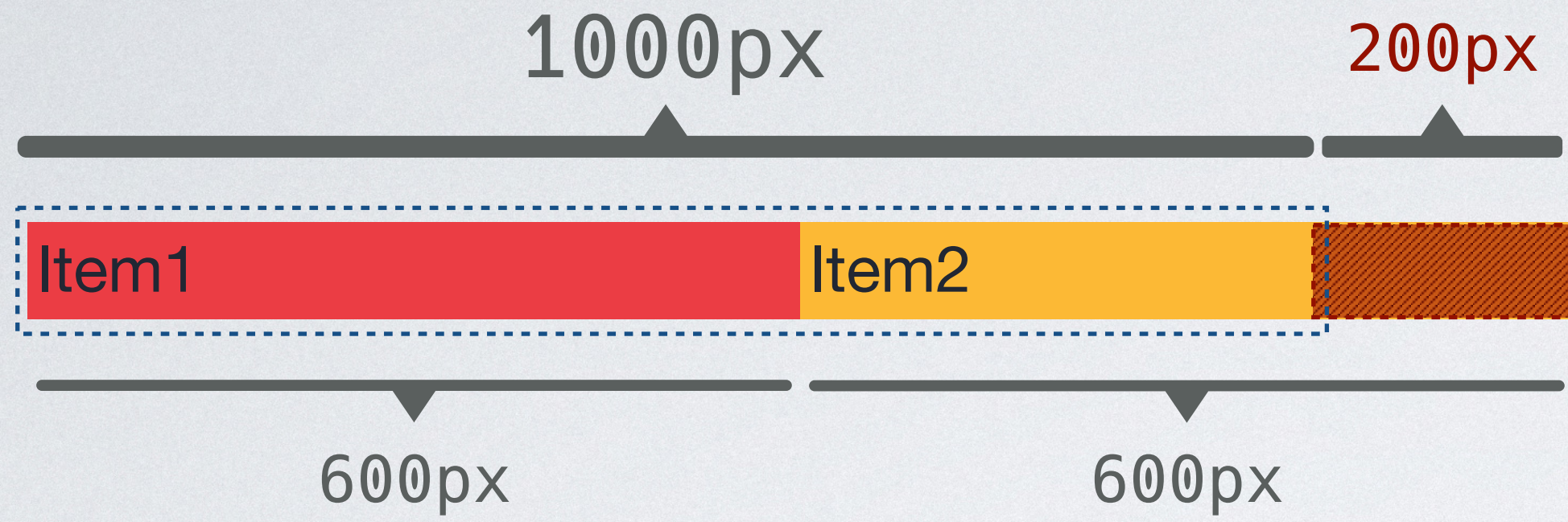
flex-shrink

number (default: 1)

Not negative

1

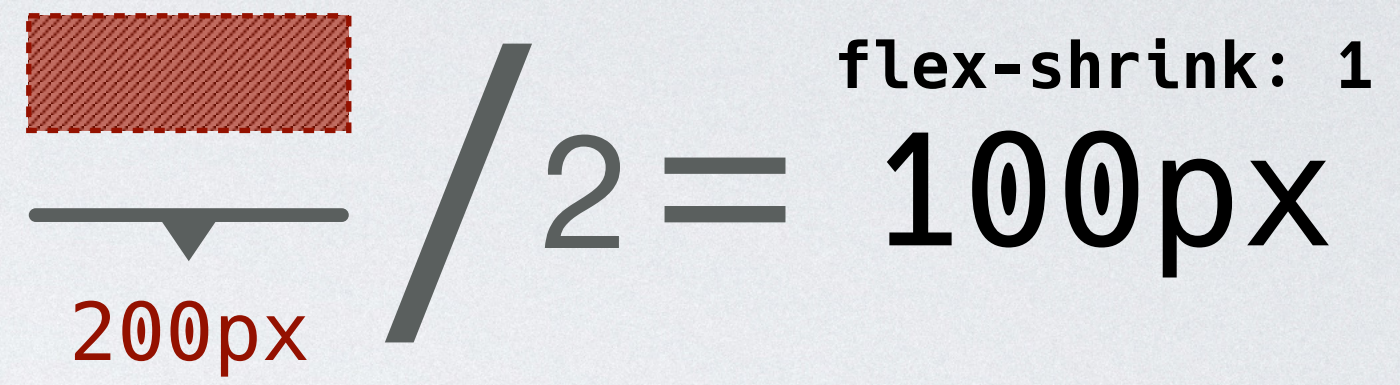
```
.i1 {flex-basis: 600px; flex-shrink: 0;}
.i2 {flex-basis: 600px; flex-shrink: 0;}
```



2

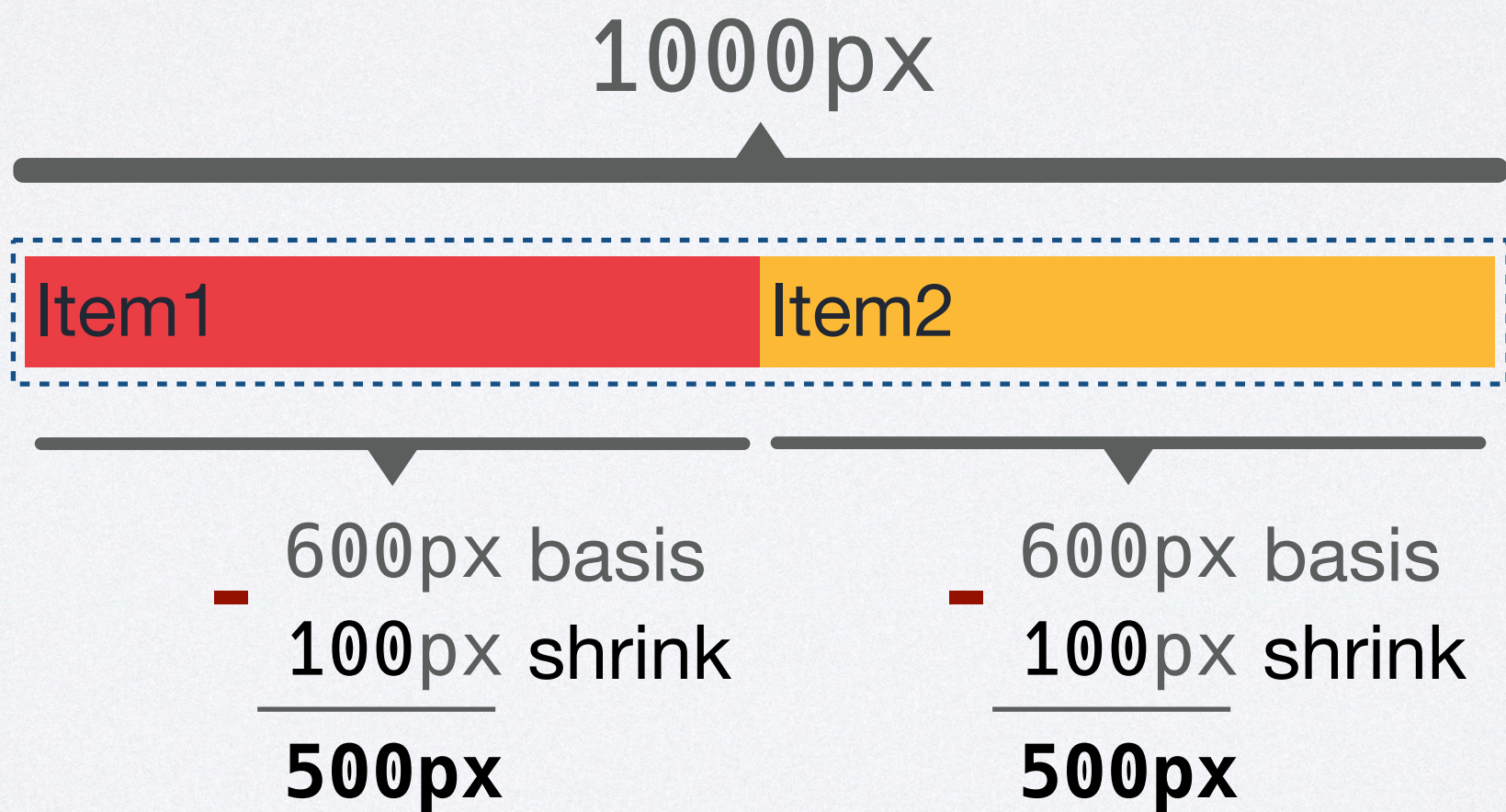
```
.i1 {flex-basis: 600px; flex-shrink: 1 /*Default*/}
.i2 {flex-basis: 600px; flex-shrink: 1 /*Default*/}
```

2



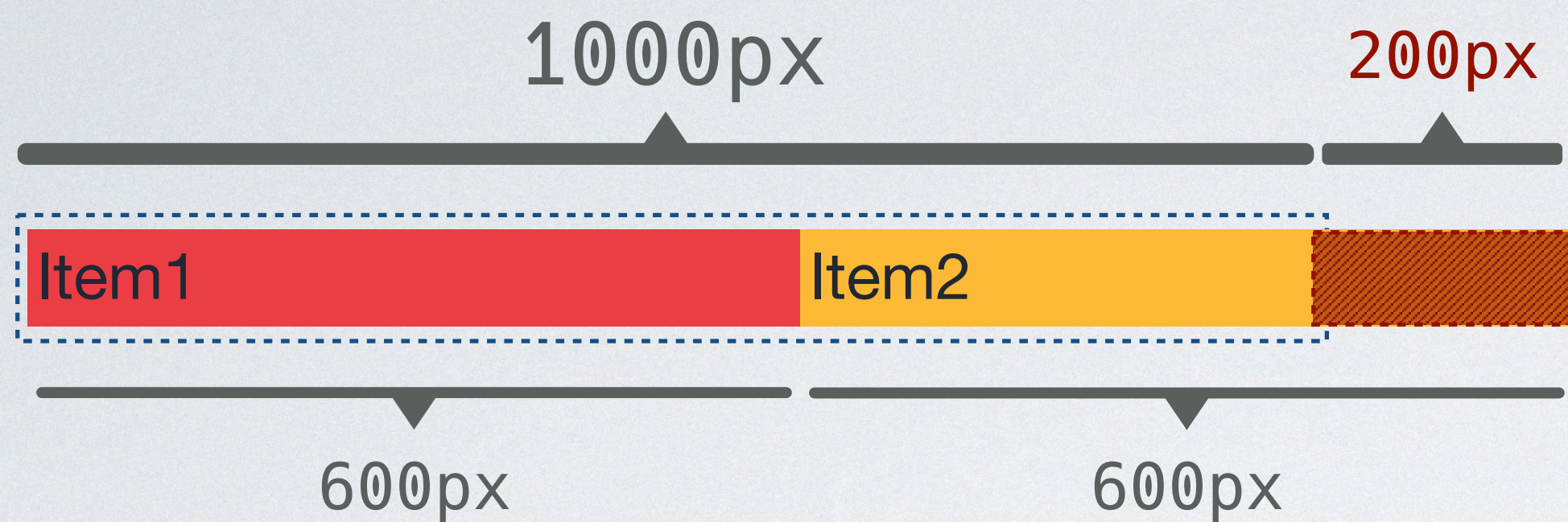
3

```
.i1 {flex-basis: 600px; flex-shrink: 1;}
.i2 {flex-basis: 600px; flex-shrink: 1;}
```



1

```
.i1 {flex-basis: 600px; flex-shrink: 0;}
.i2 {flex-basis: 600px; flex-shrink: 0;}
```



2

```
.i1 {flex-basis: 600px; flex-shrink: 1 /*Default*/}
.i2 {flex-basis: 600px; flex-shrink: 2 /*Default*/}
```

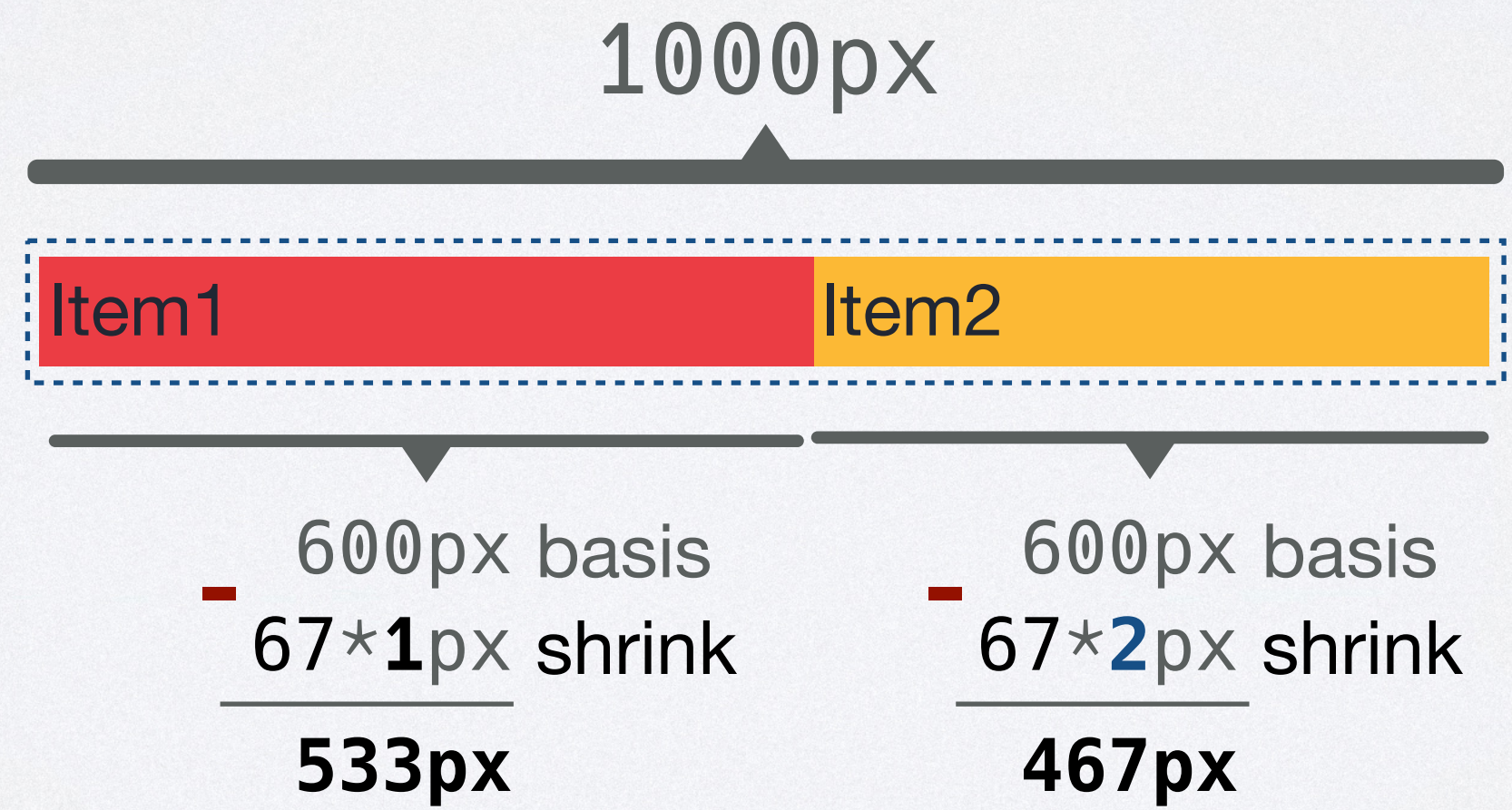
3

$$\frac{200px}{3} = 67px$$

flex-shrink: 1

3

```
.i1 {flex-basis: 600px; flex-shrink: 1;}
.i2 {flex-basis: 600px; flex-shrink: 2;}
```



FLEX-SHRINK

HTML

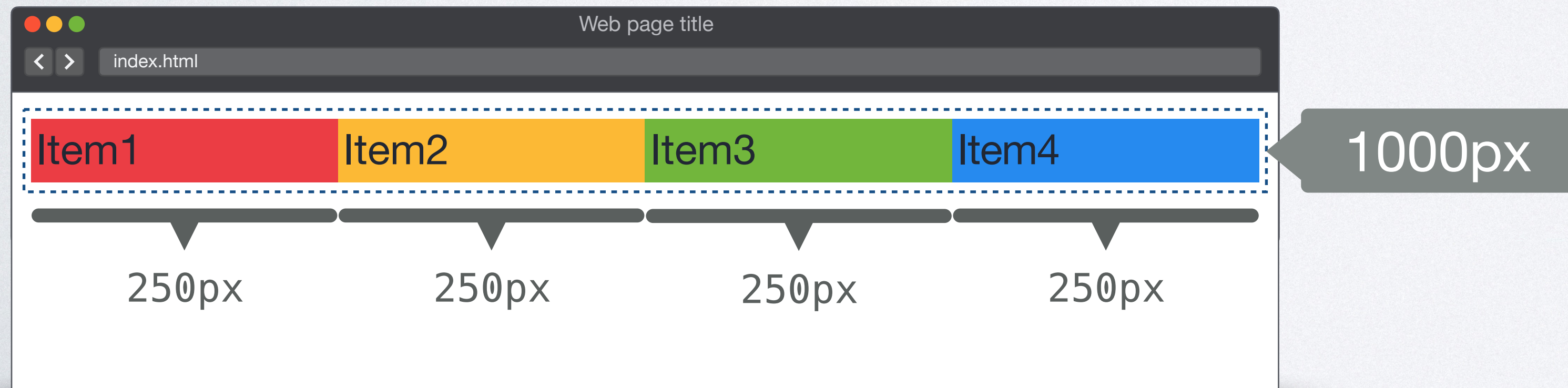
```
<body>
<div class="container">
  <div class="i1">Item1</div>
  <div class="i2">Item2</div>
  <div class="i3">Item3</div>
  <div class="i4">Item4</div>
</div>
</body>
```

CSS

```
.container {
  display: flex;
  width: 1000px;
}

.i1, .i2, .i3, .i4 {
  flex-basis: 250px;
}
```

Browser



HTML

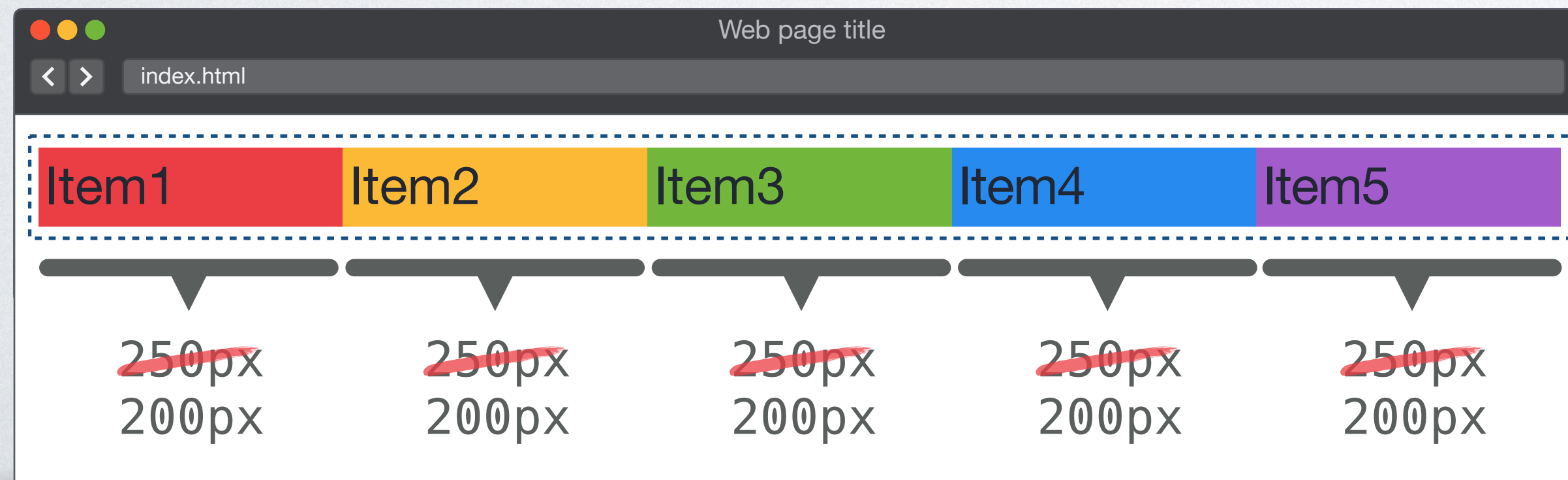
```
<body>
<div class="container">
  <div class="i1">Item1</div>
  <div class="i2">Item2</div>
  <div class="i3">Item3</div>
  <div class="i4">Item4</div>
  <div class="i5">Item5</div>
</div>
</body>
```

CSS

```
.container {
  display: flex;
  width: 1000px;
}

.i1, .i2, .i3, .i4, .i5 {
  flex-basis: 250px;
}
```

Browser



1000px

Default shrinking applied.
flex-shrink: 1

HTML

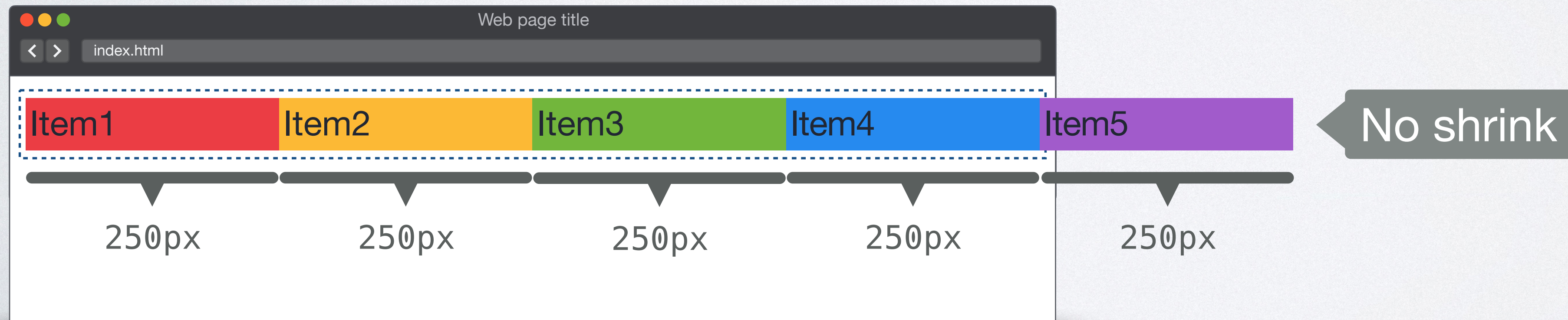
```
<body>
<div class="container">
  <div class="i1">Item1</div>
  <div class="i2">Item2</div>
  <div class="i3">Item3</div>
  <div class="i4">Item4</div>
  <div class="i5">Item5</div>
</div>
</body>
```

CSS

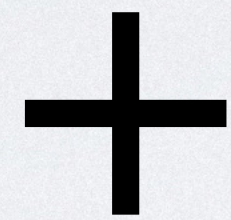
```
.container {
  display: flex;
  width: 1000px;
}

.i1, .i2, .i3, .i4, .i5 {
  flex-basis: 250px;
  flex-shrink: 0;
}
```

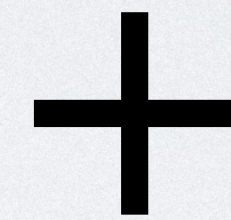
Browser



FLEX-BASIS



FLEX-GROW



FLEX-SHRINK



FLEX

flex

flex-grow flex-shrink flex-basis

none

W3C: “Authors are encouraged to control flexibility using the `flex` shorthand rather than with `flex-basis`, `flex-grow` and `flex-shrink` directly.”

flex

flex-grow

Default: **0**
If omitted: **1**

flex-shrink

Default: **1**
If omitted: **1**

flex-basis

Default: **auto**
If omitted: **0**

none

flex:

1

1

20%

=

flex-grow: 1

flex-shrink: 1

flex-basis: 20%

flex

flex-grow flex-shrink flex-basis

none

flex:

none

=

flex-grow: 0

flex-shrink: 0

flex-basis: auto

flex:

auto

=

flex-grow: 1

flex-shrink: 1

flex-basis: auto

flex



flex-grow

Default: 0
If omitted: 1

flex-shrink

Default: 1
If omitted: 1

flex-basis

Default: **auto**
If omitted: 0



none

flex: 0 0 auto



Flexulator

grow: 1

shrink: 1

basis: 100

add

1280	-	300	=	980
Container		Total Flex Basis		Remaining Space

427

426

427

FLEXBOX & GRID

— BASIS, GROW & SHRINK —



IN A ROCKET

Learn front-end development at rocket speed