



# IF.03.01 \_BASIC WEB TECHNIQUES

Creating Web Sites with HTML, CSS and Javascript



# UNIT 04

CSS Classes and Ids, Box Model, Floating, Debugging

# TODAY

- Repetition of terminology
- id and class selectors
- The css box model
- Floating elements using css
- Inspect html / css ... elements with Browser tools

# JUST LOOK AND ENJOY

- <http://www.csszengarden.com/>
- Note
  - No change to the html file
  - Only different css files are applied



# TERMINOLOGY REPETITION

- Html **elements**
  - Consist of **tags**
    - May contain **attributes**
  - May be void or not
- CSS **Rules**
  - Contain **selectors**
  - Contain **declarations**
    - Contain a **property**
    - Contain a **value**

# CSS CLASSES AND IDS



# MOTIVATION

- Consider two different tables
  - One to display personal information
  - Another to display larger data sets
  - Both to be displayed on the same page

Name	Peter Bauer
Department	Informatics
School	HTL Leonding
Country	Austria

Class	Size	Form Teacher	Average
1AHIF	32	R. Brenn	2.5
1BHIF	32	F. Auernig	1.9
1CHIF	32	D. Klewein	2.2
1DHIF	32	K. Lugmayr	2.0
2AHIF	32	E. Keplinger	3.5
2BHIF	32	E. Rumetshofer	3.1
2CHIF	32	R. Denkmair	2.8
3AHIF	20	B. Schröder	2.3
3BHIF	32	G. Köck	1.5
3CHIF	32	F. Jakob	2.7
4AHIF	20	R. Traunmüller	3.3
4BHIF	25	M. Kerschner	2.9
5AHIF	22	P. Bauer	1.8
5BHIF	20	H. Lackinger	2.1
5CHIF	20	M. Bucek	2.4

Maybe even  
more than once

# MOTIVATION

Name	Peter Bauer
Department	Informatics
School	HTL Leonding
Country	Austria

personal-information

- How to address these tables?

- Inline css

A thing we want to avoid

- Css class

data-table

Class	Size	Form Teacher	Average
1AHIF	32	R. Brenn	2.5
1BHIF	32	F. Auernig	1.9
1CHIF	32	D. Klewein	2.2
1DHIF	32	K. Lugmayr	2.0
2AHIF	32	E. Keplinger	3.5
2BHIF	32	E. Rumetshofer	3.1
2CHIF	32	R. Denkmair	2.8
3AHIF	20	B. Schröder	2.3
3BHIF	32	G. Köck	1.5
3CHIF	32	F. Jakob	2.7
4AHIF	20	R. Traunmüller	3.3
4BHIF	25	M. Kerschner	2.9
5AHIF	22	P. Bauer	1.8
5BHIF	20	H. Lackinger	2.1
5AHIF	20	M. Bucek	2.4



# HOWTO DEFINE A CLASS

```
<table class="personal-information">
```

Class attribute

Class selector

```
table.personal-information {  
    border: 1px solid grey;  
}
```

# WHEN TO USE A CLASS?

- Make an Html element addressable from an internal/external Css
- The Html element may appear more often than once on the page



# EXAMPLES FOR CLASSES

- Elements of a specific type

- table.personal\_data, p.message

`<p class = "message is-error">`

- Elements in a specific state

- p.message.is-error

Especially in WebApps this  
will become a topic

- Make general containers (`<div>`, `<span>`) more specific

- div.avatar, div.scene

# HOW TO DEFINE AN ID

```
<body id="mainpage">
```

Id attribute

Id selector

```
body#mainpage {
```

```
    background: yellow;
```

```
}
```



# WHEN TO USE AN ID?

- Make an Html element addressable from an internal/external Css
- The Html element may appear only once on the page

# EXAMPLES FOR IDS

- Make pages more specific
  - `body#main-page`, `body#team`
- Special elements only occurring once on a page
  - `article#teaser`



# CLASSES AND IDS CAN BE DEFINED MORE GENERAL

- You may omit the html selector
- `.error-message { border: 1px solid red; }`
  - All elements of class `error-message`
- `#mainpage { background: yellow; }`
  - All elements with id `mainpage`

# CSS BOX MODEL



# BLOCK AND INLINE ELEMENTS

# BLOCK AND INLINE ELEMENTS

- Block elements take an extra line
- article, section, p, h1-6, div



# BLOCK AND INLINE ELEMENTS

- Block elements take an extra line
- article, section, p, h1-6, div

```
<h2>The Fun is There</h2>  
<p>If we start talking  
about block level  
elements. Block level  
elements use an extra  
line if they begin. Like  
this p element here.</p>
```

## **The Fun is There**

If we start talking about block level elements. Block level elements use an extra line if they begin. Like this p element here.

# BLOCK AND INLINE ELEMENTS

- Block elements take an extra line
  - article, section, p, h1-6, div
- Inline elements are in the “normal” text flow
  - img, strong, em, img, span



# BLOCK AND INLINE ELEMENTS

- Block elements take an extra line
  - article, section, p, h1-6, div
- Inline elements are in the “normal” text flow
  - img, strong, em, span

```
<p>If we start talking  
about <em> inline</em>  
elements.  
<strong>Inline</strong>  
elements use an extra  
line if they begin. Like  
the em and strong  
elements above.</p>
```

If we start talking about *inline* elements. **Inline** elements use an extra line if they begin. Like the em and strong elements above.

# BLOCK ELEMENTS



# BLOCK ELEMENTS

- Every Html Element can be considered as a box

# BLOCK ELEMENTS

- Every Html Element can be considered as a box
- `<h1>CSS Box Model</h1>`



# BLOCK ELEMENTS

- Every Html Element can be considered as a box
- `<h1>CSS Box Model</h1>`

CSS Box Model

# BLOCK ELEMENTS

- Every Html Element can be considered as a box
- `<h1>CSS Box Model</h1>`

```
h1 {
```

```
border: 20px solid grey;
```

CSS Box Model

```
}
```



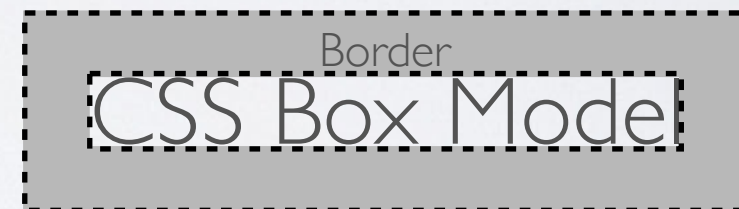
# BLOCK ELEMENTS

- Every Html Element can be considered as a box
- `<h1>CSS Box Model</h1>`

```
h1 {
```

```
    border: 20px solid grey;
```

```
}
```



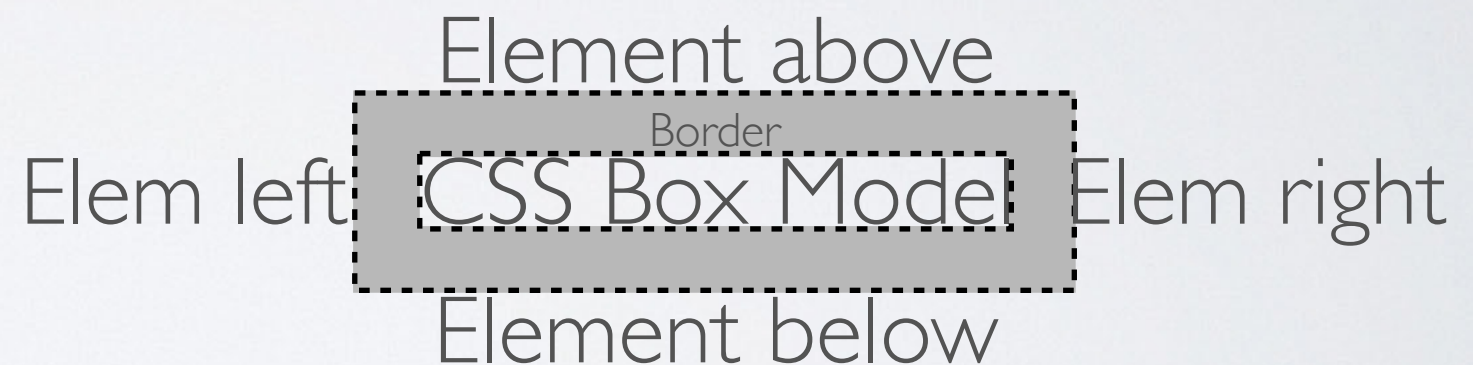
# BLOCK ELEMENTS

- Every Html Element can be considered as a box
- `<h1>CSS Box Model</h1>`

```
h1 {
```

```
border: 20px solid grey;
```

```
}
```





# BLOCK ELEMENTS

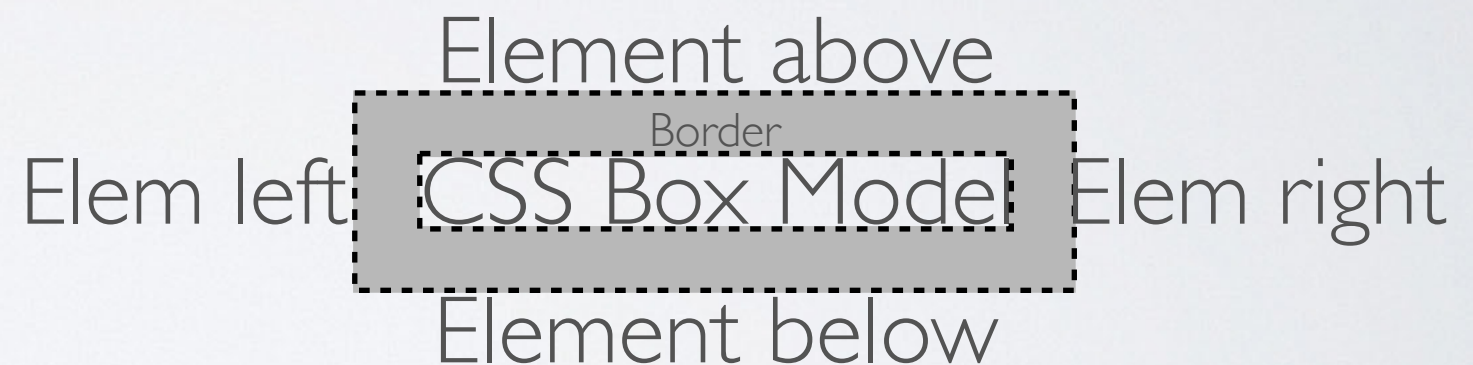
- Every Html Element can be considered as a box
- `<h1>CSS Box Model</h1>`

```
h1 {
```

```
    border: 20px solid grey;
```

```
    padding: 100px;
```

```
}
```



# BLOCK ELEMENTS

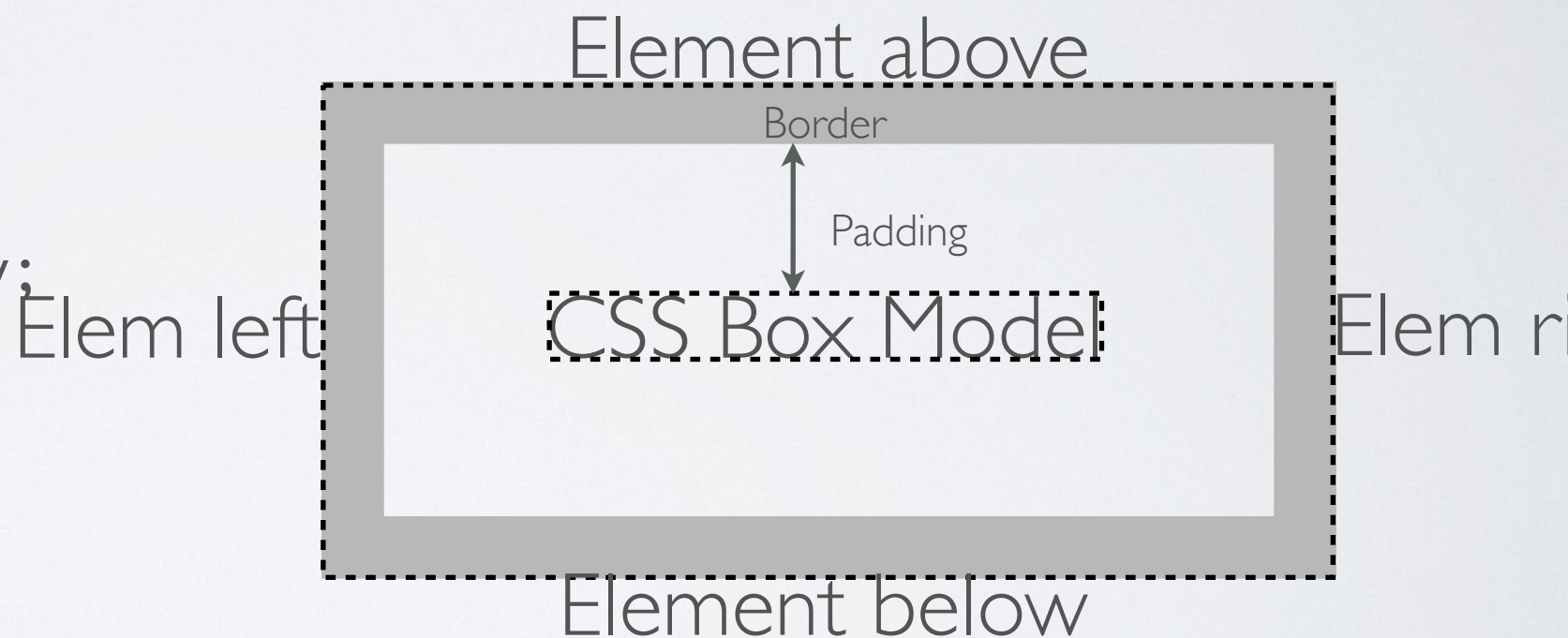
- Every Html Element can be considered as a box
- `<h1>CSS Box Model</h1>`

```
h1 {
```

```
border: 20px solid grey;
```

```
padding: 100px;
```

```
}
```

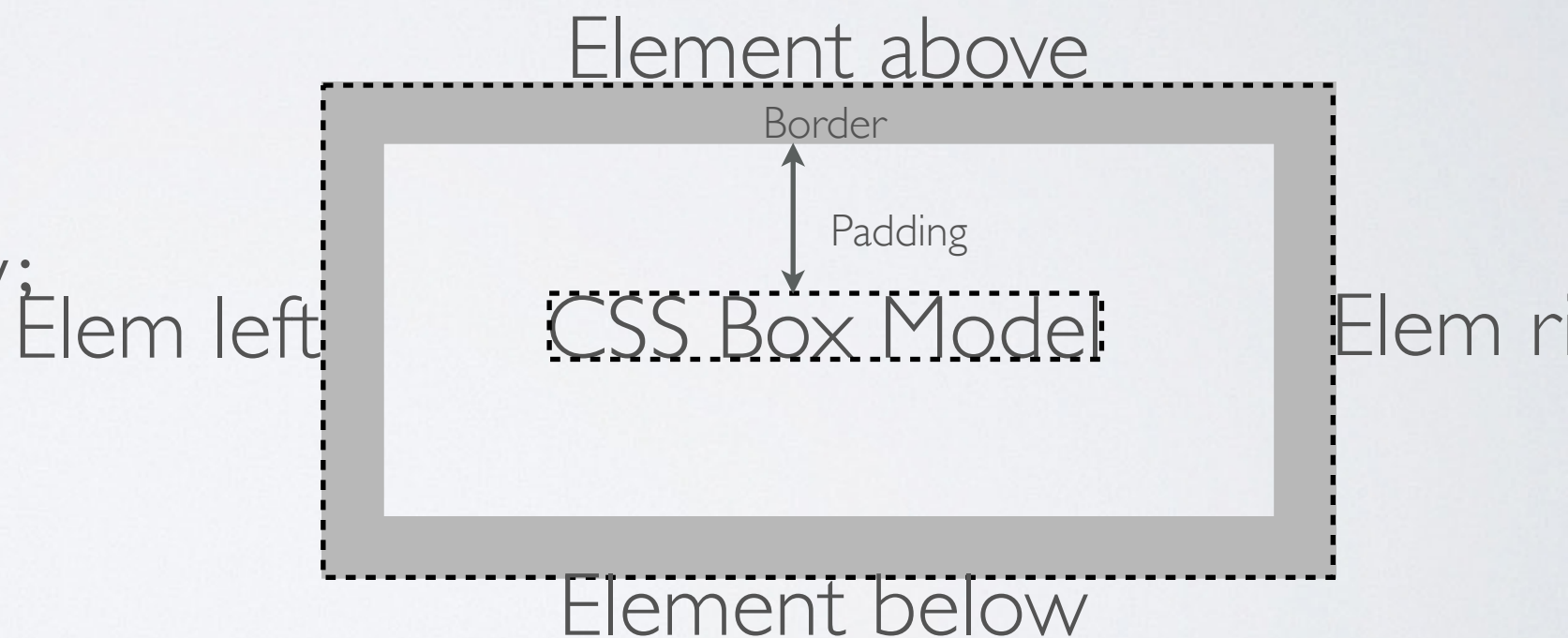




# BLOCK ELEMENTS

- Every Html Element can be considered as a box
- `<h1>CSS Box Model</h1>`

```
h1 {  
  
    border: 20px solid grey;  
  
    padding: 100px;  
  
    margin: 100px;  
  
}
```



# BLOCK ELEMENTS

- Every Html Element can be considered as a box

- `<h1>CSS Box Model</h1>`

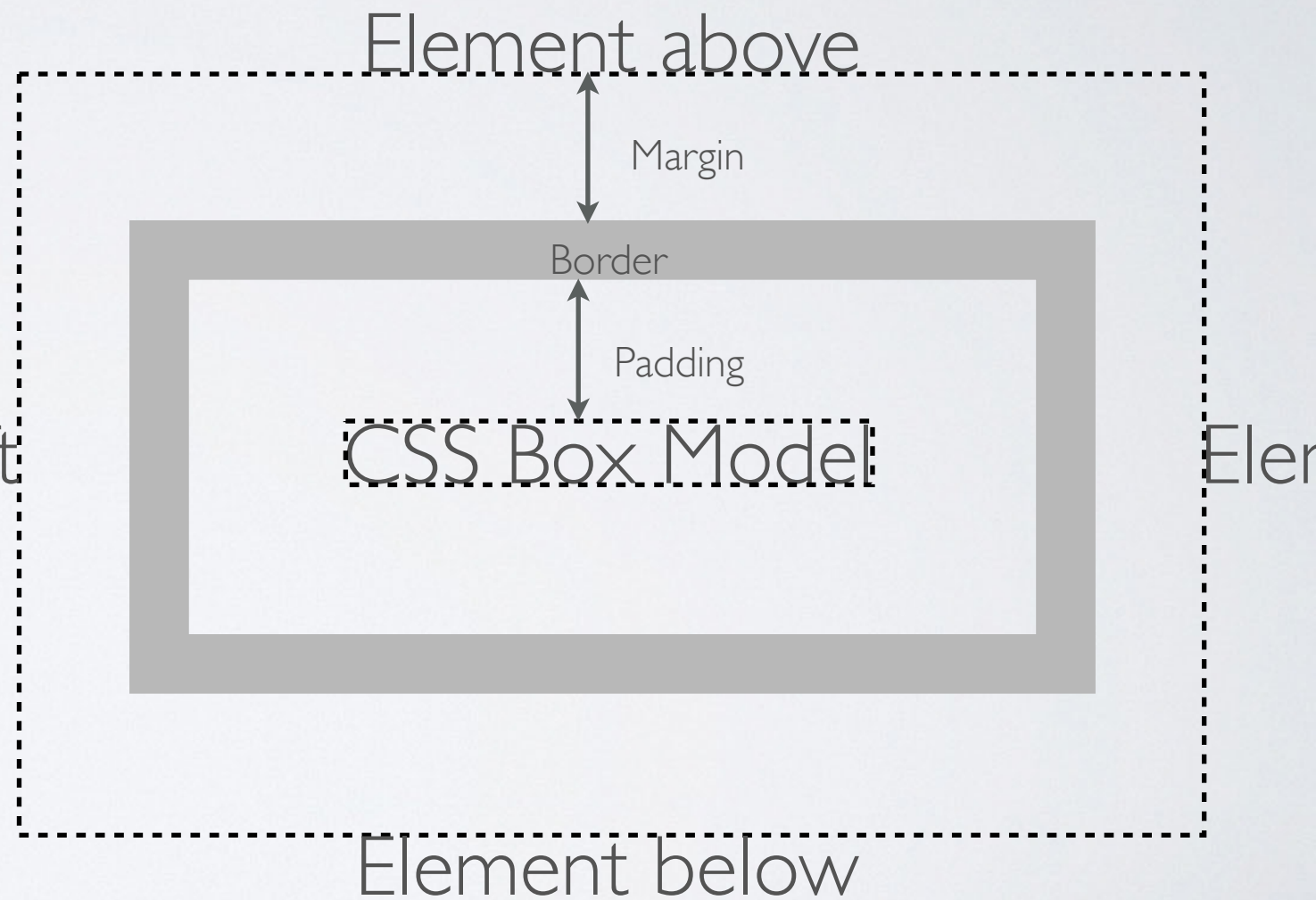
```
h1 {
```

```
  border: 20px solid grey;
```

```
  padding: 100px;
```

```
  margin: 100px;
```

```
}
```





# BLOCK ELEMENTS

- Every Html Element can be considered as a box

- `<h1>CSS Box Model</h1>`

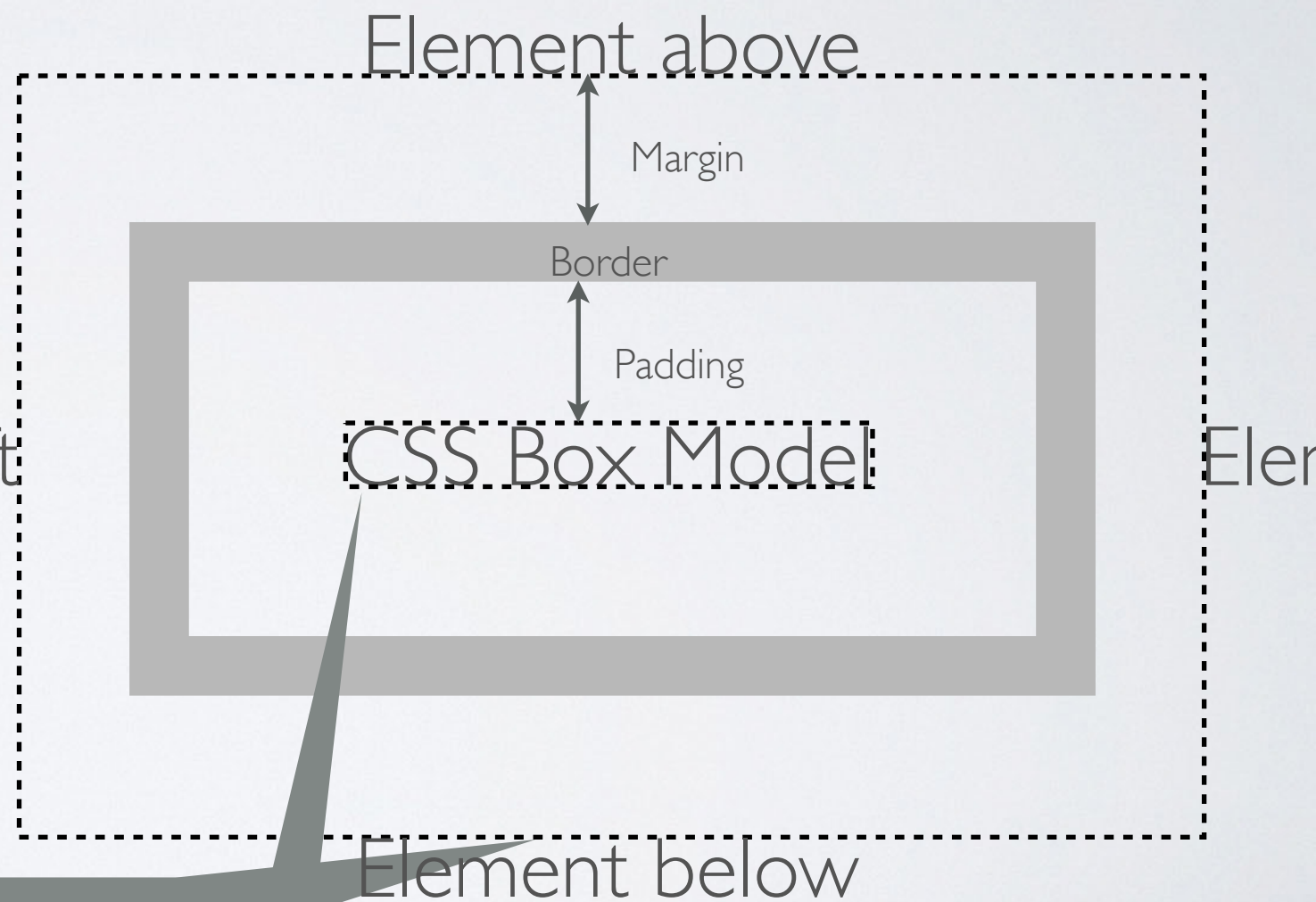
```
h1 {
```

```
border: 20px solid grey;
```

```
padding: 100px;
```

```
margin: 100px;
```

```
}
```



Note: The dotted lines are imaginary lines

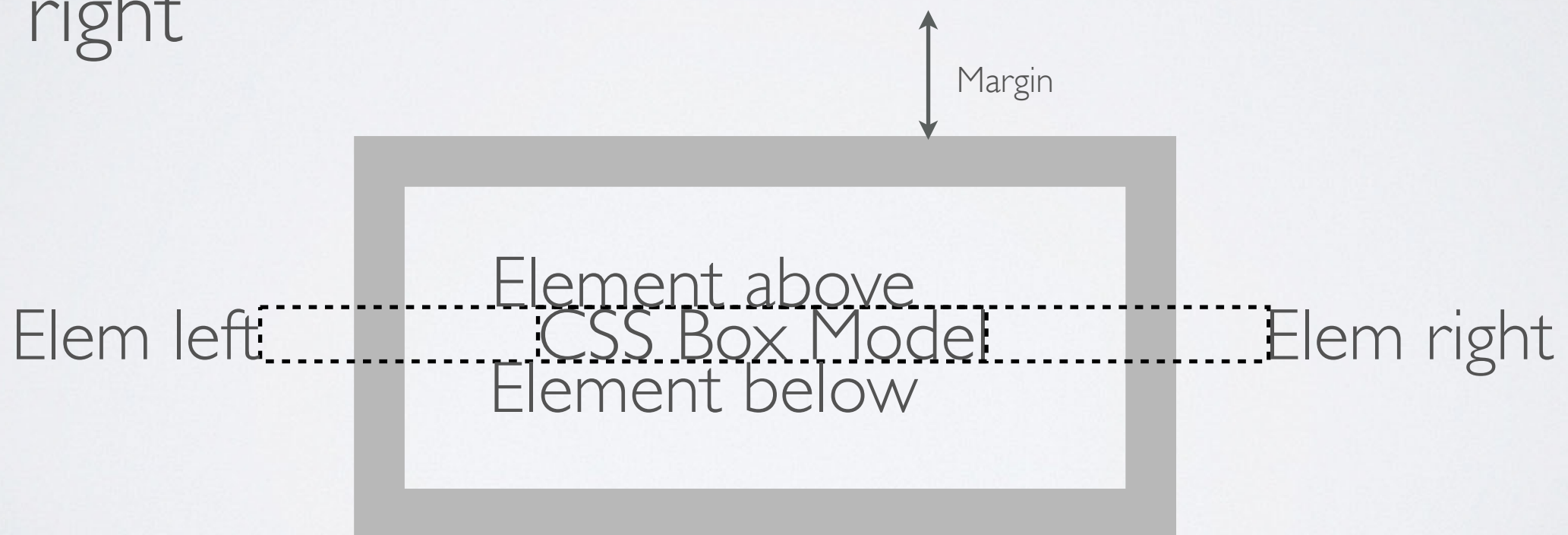
# INLINE ELEMENTS

- In principal the same
- Surrounding elements move only to the left and to the right



# INLINE ELEMENTS

- In principal the same
- Surrounding elements move only to the left and to the right



# FLOATING ELEMENTS



# WHY?

- All Html elements are in a “text flow”
- That means they all come one after the next
- Sometimes this is not desirable
- We want to pull elements out of the text flow

## Personal Data

Name: Peter Bauer  
Date of Birth: March 23, 1967  
Address: Mühlkreisbahnstr. 5, 4040 Linz  
Phone: +43 676 6173320  
e-mail: p.bauer@htl-leonding.ac.at  
Family Status: Married, 2 children (1999 and 2002)  
Citizenship: Austrian



# WHY?

- All Html elements are in a “text flow”
- That means they all come one after the next
- Sometimes this is not desirable
- We want to pull elements out of the text flow
- They should float

## Personal Data

Name: Peter Bauer  
Date of Birth: March 23, 1967  
Address: Mühlkreisbahnstr. 5, 4040 Linz  
Phone: +43 676 6173320  
e-mail: p.bauer@htl-leonding.ac.at  
Family Status: Married, 2 children (1999 and 2002)  
Citizenship: Austrian



## Personal Data

Name: Peter Bauer  
Date of Birth: March 23, 1967  
Address: Mühlkreisbahnstr. 5, 4040 Linz  
Phone: +43 676 6173320  
e-mail: p.bauer@htl-leonding.ac.at  
Family Status: Married, 2 children (1999 and 2002)  
Citizenship: Austrian





# FLOATING IN ACTION

```

```



# FLOATING IN ACTION

```

```

```
<table>
```

```
...
```

```
<td>e-mail</td>...
```

```
</table>
```



Name:	Peter Bauer
e-mail:	<a href="mailto:p.bauer@htl-leonding.ac.at">p.bauer@htl-leonding.ac.at</a>
...	...
Citizenship:	Austrian



# FLOATING IN ACTION

```

```

```
<table>
```

```
...
```

```
<td>e-mail</td>...
```

```
</table>
```

```
img {float: right;}
```

Name:	Peter Bauer
e-mail:	<a href="mailto:p.bauer@htl-leonding.ac.at">p.bauer@htl-leonding.ac.at</a>
...	...
Citizenship:	Austrian



# FLOATING IN ACTION

```

```

```
<table>
```

```
...
```

```
<td>e-mail</td>...
```

```
</table>
```

```
img {float:left;}
```



Name:	Peter Bauer
e-mail:	<a href="mailto:p.bauer@htl-leonding.ac.at">p.bauer@htl-leonding.ac.at</a>
...	...
Citizenship:	Austrian



# CLEAR FLOATING

```
<!DOCTYPE html>
<html>
<head>
<style>
img {
  float: left;
}

p.clear {
  clear: left;
}
</style>
</head>
<body>

<h1>The clear Property</h1>



<p class="clear">This is some text. This is some text. This is some text. This is some text. This
is some text. This is some text.</p>
<p><strong>Remove the "clear" class to see the effect.</strong></p>

</body>
</html>
```

## The clear Property



This is some text. This is some text. This is some text. This is some text. This is some text. This is some text.

**Remove the "clear" class to see the effect.**

# CLEAR FLOATING

```
<!DOCTYPE html>
<html>
<head>
<style>
img {
  float: left;
}

p.clear {
  clear: left;
}
</style>
</head>
<body>

<h1>The clear Property</h1>



<p>This is some text. This is some text. This is some text. This is some text. This is some text.
This is some text.</p>
<p><strong>Remove the "clear" class to see the effect.</strong></p>

</body>
</html>
```

## The clear Property



This is some text. This is some text. This is some text. This is some text. This is some text. This is some text.

Remove the "clear" class to see the effect.



DEBUGGING

# INSPECT ELEMENT

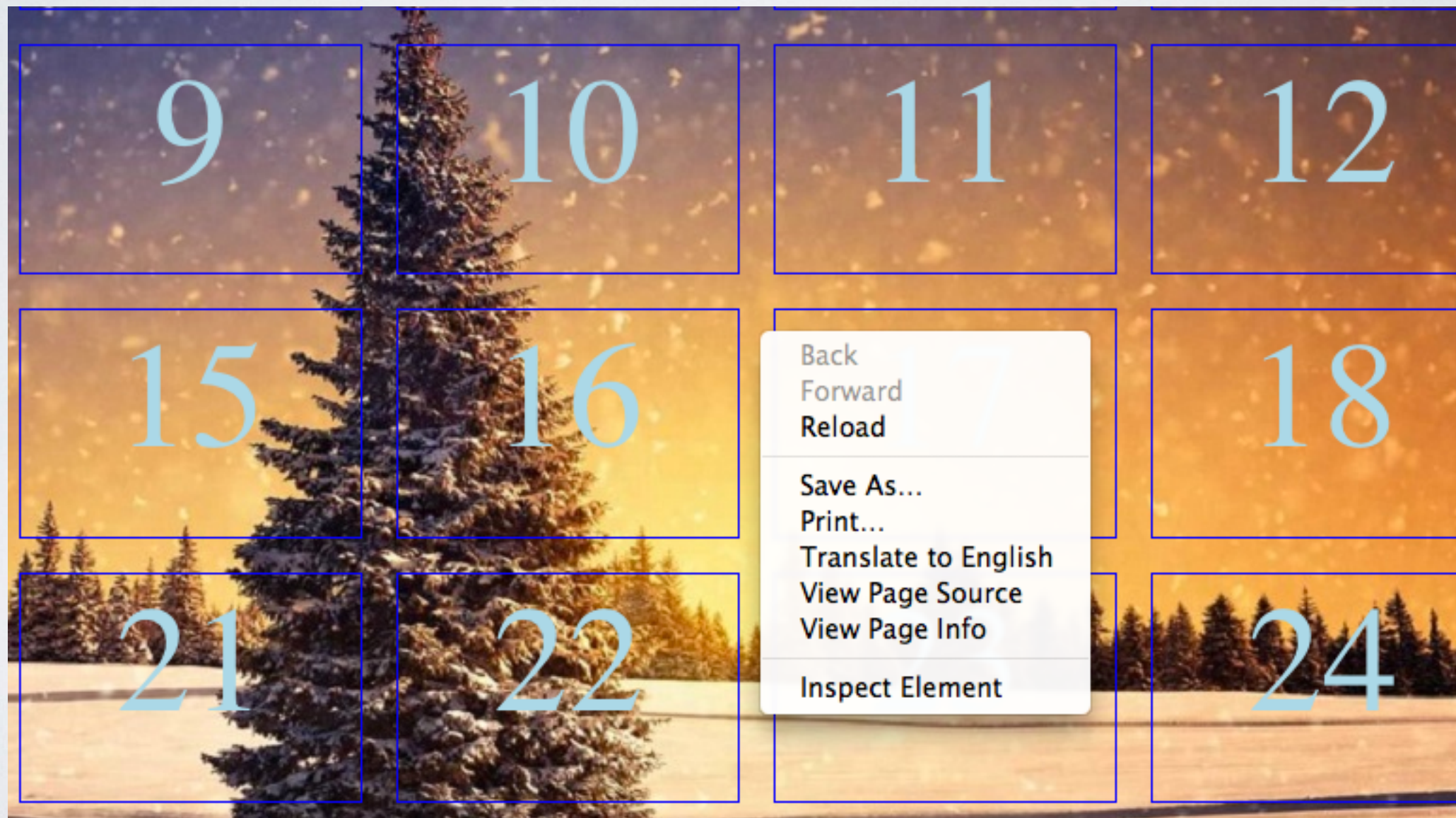
- It is useful to look into the Html elements as the browser sees it
- “Debugging”
  - What size is this element?
  - Which Css rule does apply to this element?
  - How would it be if a Css rule wouldn't be there?
  - ...



# THE BROWSER HELPS

- Nowadays all browsers support this kind of debugging
- Firefox was the first with the plug-in “Firebug”

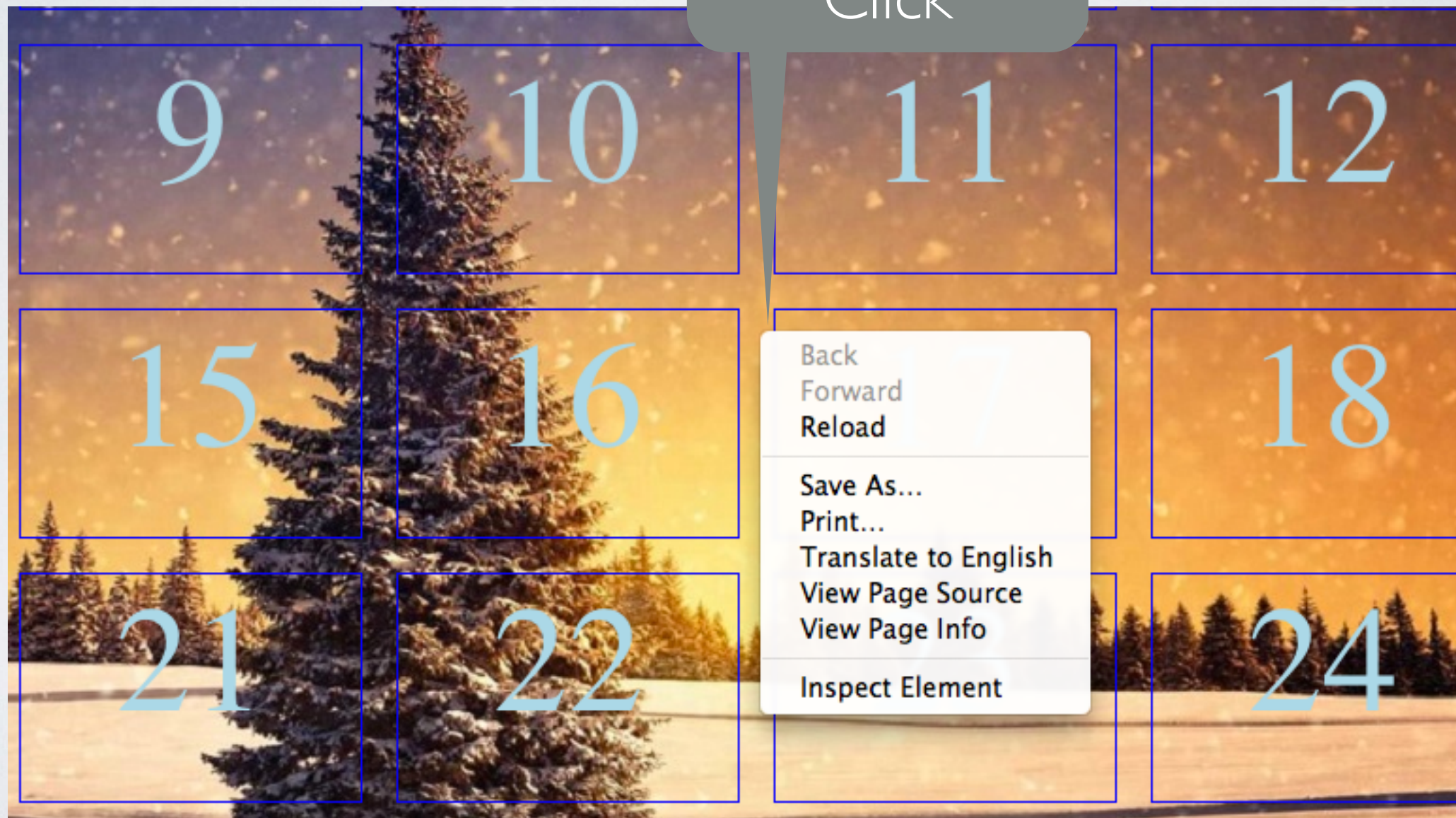
# INSPECT ELEMENT – START





# INSPECT ELEMENT – START

Right Mouse  
Click





# INSPECT ELEMENT – START

