



# EDTA, iText and INBATEK Conference

Bangkok, July 27, 2017

# Three major ways to create PDF

## ≡ From scratch using iText objects:

- ≡ Paragraph, Text, Table, Cell, List,...

- ≡ Advantage: everything is programmable, hence configurable just the way you want it,

- ≡ Disadvantage: you need to program everything; small changes need programming.

## ≡ Using an existing template:

- ≡ AcroForm technology: fast, easy, but not dynamic (all fields have fixed positions),

- ≡ XFA technology: dynamic, filling is easy, but form creation is complex, deprecated in PDF 2.0!

## ≡ By converting HTML + CSS to PDF:

- ≡ Everyone knows HTML! Everyone knows CSS!

- ≡ Why would you create a template in another (proprietary?) format?

- ≡ Create content in HTML, convert to PDF with pdfHTML.



# CSS examples

- No CSS,
- Inline CSS,
- Header CSS,
- External CSS.

# No CSS: HTML + pdfHTML code

```
<html>
<head>
<title>Colossal</title>
<meta name="description" content="Gloria is an out-of-work party girl forced to leave her life in New
</head>
<body>

<h1>Colossal (2016)</h1>
<i>Directed by Nacho Vigalondo</i>
<div>Gloria is an out-of-work party girl forced to leave her life in New York City, and move back hom
<font size="-1">Read more about this movie on <a href="www.imdb.com/title/tt4680182">IMDB</a></font>
</body>
</html>
```

```
public void createPdf(String src, String dest) throws IOException {
    HtmlConverter.convertToPdf(new File(src), new File(dest));
}
```

# No CSS: result



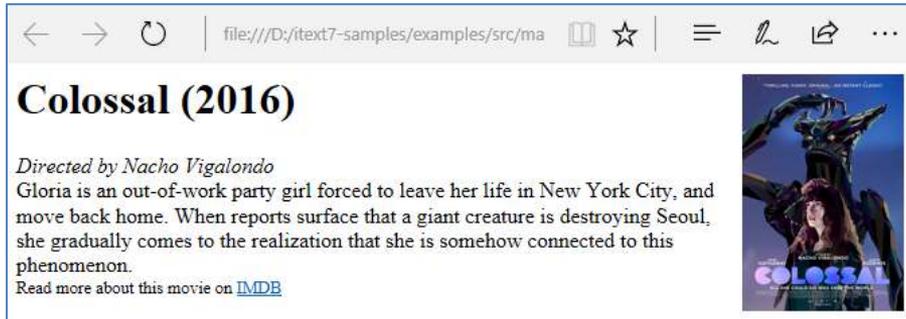
# Inline CSS: HTML + pdfHTML code

```
<html>
<head>
<title>Colossal</title>
<meta name="description" content="Gloria is an out-of-work party girl forced to leave her life in New York City, and
</head>
<body>

<h1>Colossal (2016)</h1>
<div style="font-style: italic">Directed by Nacho Vigalondo</div>
<div>Gloria is an out-of-work party girl forced to leave her life in New York City, and move back home. When reports
<div style="font-size: 0.8em">Read more about this movie on <a href="www.imdb.com/title/tt4680182">IMDB</a></font>
</body>
</html>
```

```
public void createPdf(String src, String dest) throws IOException {
    HtmlConverter.convertToPdf(new File(src), new File(dest));
}
```

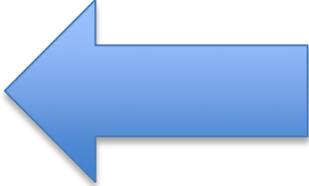
# Inline CSS: result



# Header CSS: HTML + pdfHTML code

```
<html>
<head>
  <title>Colossal</title>
  <meta name="description" content="Gloria is an out-of-work party girl forced to leave her life in New Yc
</style>
.poster { width: 120px;float: right; }
.director { font-style: italic; }
.description { font-family: serif; }
.imdb { font-size: 0.8em; }
a { color: red; }
</style>
</head>
<body>

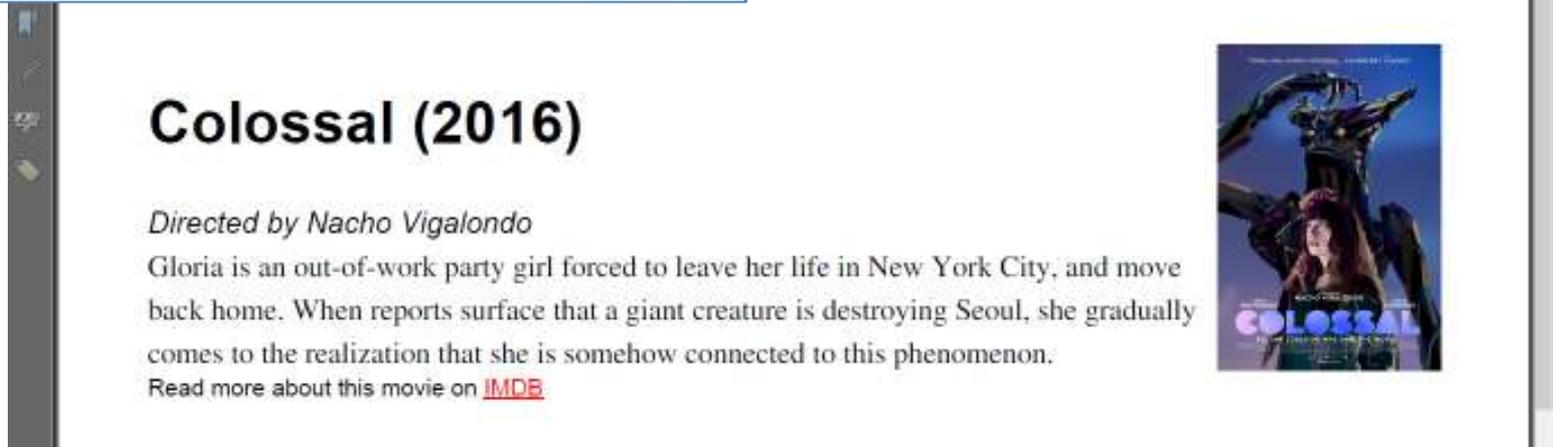
<h1>Colossal (2016)</h1>
<div class="director">Directed by Nacho Vigalondo</div>
<div class="description">Gloria is an out-of-work party girl forced to leave her life in New York City,
<div class="imdb">Read more about this movie on <a href="www.imdb.com/title/tt4680182">IMDB</a></div>
</body>
</html>
```



```
public void createPdf(String src, String dest) throws IOException {
    HtmlConverter.convertToPdf(new File(src), new File(dest));
}
```



# Header CSS: result



# External CSS: HTML + pdfHTML code

```
<html>
<head>
<title>Colossal</title>
<meta name="description" content="Gloria is an out-of-work party girl forced to leave her life in New Yo:
<link rel="stylesheet" type="text/css" href="movie.css">
</head>
<body>

<h1>Colossal (2016)</h1>
<div class="director">Directed by Nacho Vigalondo</div>
<div class="description">Gloria is an out-of-work party girl forced to leave her life in New York City, :
<div class="imdb">Read more about this movie on <a href="www.imdb.com/title/tt4680182">IMDB</a></div>
</body>
</html>
```

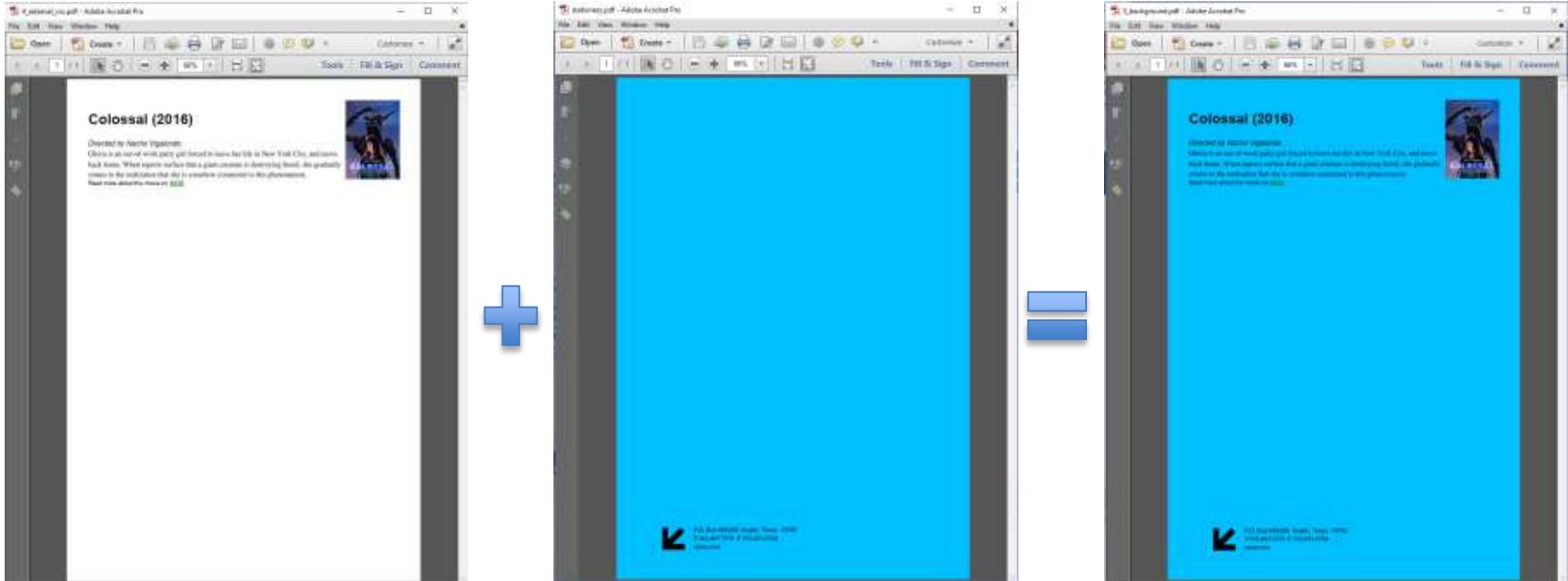
```
.poster {
width: 120px;
float: right;
}
.director {
font-style: italic;
}
.description {
font-family: serif;
}
.imdb {
font-size: 0.8em;
}
a {
color: green;
}
```

```
public void createPdf(String src, String dest) throws IOException {
    HtmlConverter.convertToPdf(new File(src), new File(dest));
}
```

# External CSS: result



# Adding a background



# Background

```
class Background implements IEventHandler {
    PdfXObject stationery;

    public Background(PdfDocument pdf, String src) throws IOException {
        PdfDocument template = new PdfDocument(new PdfReader(src));
        PdfPage page = template.getPage(1);
        stationery = page.copyAsFormXObject(pdf);
        template.close();
    }

    @Override
    public void handleEvent(Event event) {
        PdfDocumentEvent docEvent = (PdfDocumentEvent) event;
        PdfDocument pdf = docEvent.getDocument();
        PdfPage page = docEvent.getPage();
        PdfCanvas canvas = new PdfCanvas(page.newContentStreamBefore(), page.getResources(), pdf);
        canvas.addXObject(stationery, 0, 0);
    }
}
```

# Use in the context of pdfHTML

- ≡ Other `convertToPdf()` method: `FileInputStream` and `PdfDocument`
- ≡ Why do we need `ConverterProperties`?

```
public void createPdf(String src, String stationery, String dest) throws IOException {  
    PdfWriter writer = new PdfWriter(dest);  
    PdfDocument pdf = new PdfDocument(writer);  
    IEventHandler handler = new Background(pdf, stationery);  
    pdf.addEventHandler(PdfDocumentEvent.START_PAGE, handler);  
    ConverterProperties properties = new ConverterProperties();  
    properties.setBaseUri(new File(src).getParentFile().getAbsolutePath());  
    HtmlConverter.convertToPdf(new FileInputStream(src), pdf, properties);  
}
```



# Use case: invoices

For every customer you need:

- Raw data that can be converted to simple HTML
- A customer-provided CSS to apply styles
- A customer-provided PDF as company stationery

# ZUGFeRD examples

The screenshot shows a web browser window with the URL `developers.itextpdf.com/content/zugferd-future-invoicing/6-creating-html-invoices`. The browser displays an invoice for 'Das Company' with the following details:

**INVOICE I/00004**  
2015-04-01

**From:** Das Company  
ZUG Business Center  
Highway 1  
BE-9000 Ghent  
FC: 20111340209  
VA: BE123456789

**To:** Sommer, Bill  
362 - 20th Ave.  
BE-9000 Ghent

#	Product	Unit	Qty.	Sub.	Tax%	Tax	Total	Curr.
1.	Running jersey	8.00	9.00	72.00	21.00%	15.12	87.12	EUR
2.	Golf polo	8.00	1.00	8.00	21.00%	1.68	9.68	EUR
3.	Threadmill	600.00	2.00	1200.00	21.00%	252.00	1452.00	EUR
4.	Pro steel dartboard	25.00	2.00	50.00	21.00%	10.50	60.50	EUR
5.	My First Cookbook	17.00	1.00	17.00	6.00%	1.02	18.02	EUR
6.	Golf kit	80.00	2.00	160.00	21.00%	33.60	193.60	EUR

Tax	%	Base amount:	Tax amount:	Total	Curr.
VAT	6.00	16.04	0.96	17.00	EUR
VAT	21.00	1231.40	258.60	1489.99	EUR
Grand total:		1247.44	259.56	1507.00	EUR

Please wire the amount due to our bank account using the following reference: 005000004

Bank	BIC ID	IBAN Number
KBC	KREDDEBB	BE 41 7360 0661 9710
BNP Paribas	GERABEBB	BE 56 0015 4298 7888

Figure 6.2: HTML + CSS version of an invoice

Now we're getting somewhere, aren't we? In our XSL, we've added some `id` and some `class` attributes, so that we can refer to these elements from a simple CSS file:

The screenshot shows a PDF viewer window with the URL `developers.itextpdf.com/content/zugferd-future-invoicing/7-creating-pdf-invoices-comfort`. The window title is 'The final result'. The PDF content is identical to the HTML invoice shown in Figure 6.2, but rendered as a professional PDF document with a clean layout and proper styling.

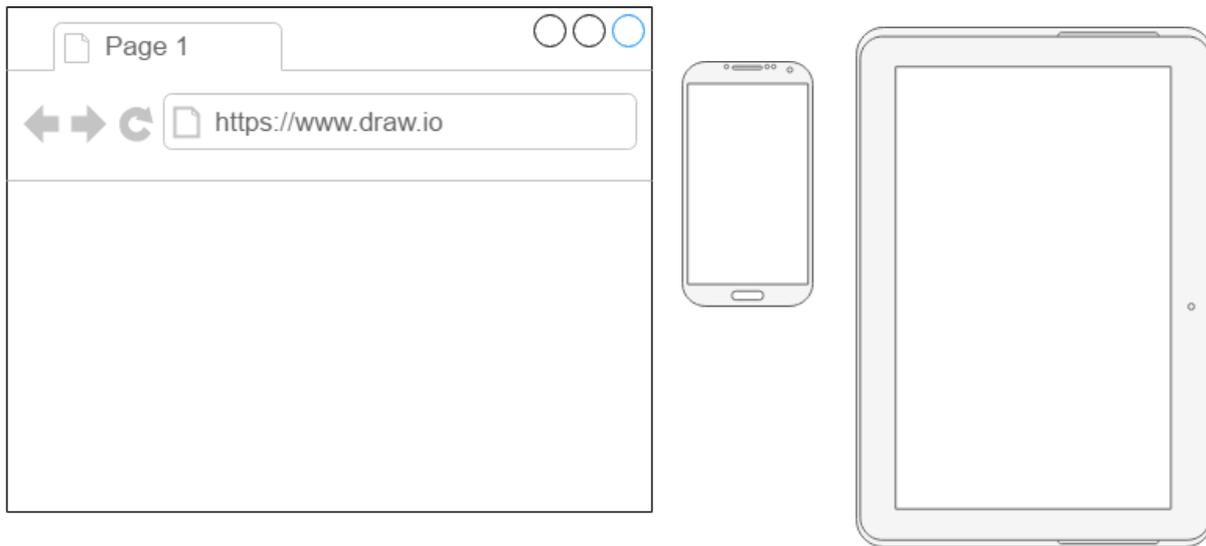
Figure 7.1: a ZUGFeRD invoice created from HTML



# ☰ Responsiveness: HTML vs PDF

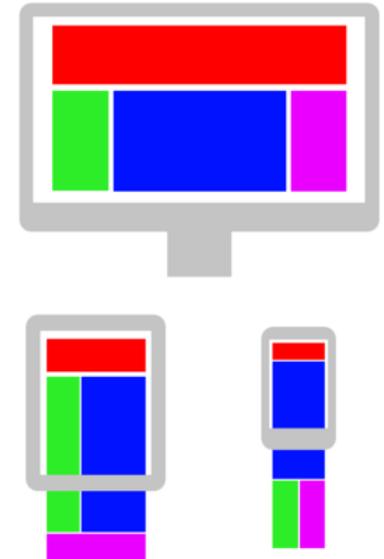
# HTML and different devices

☰ When creating an HTML + CSS file, you don't always know how it will be viewed:



# HTML and different devices

- ≡ The content can be rendered differently based on the device:
  - ≡ Header, main, left and right pane can be reorganized based on available space
  - ≡ Navigation can be different on different devices
  - ≡ Orientation can change (e.g. turn tablet from portrait to landscape)
- ≡ You can change the rendering based on Media Queries
  - ≡ Change font, widths, image size, positions,...
    - Desktop screens: horizontal placement, smaller font-size,...
    - Mobile screen: larger font size, vertical flow,...
    - Print: hide interactive and animated features, use less color,...

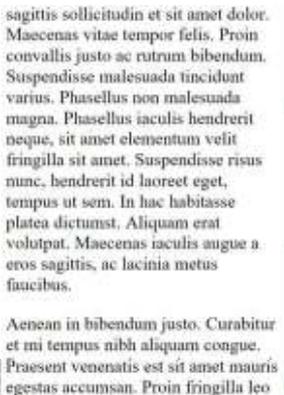
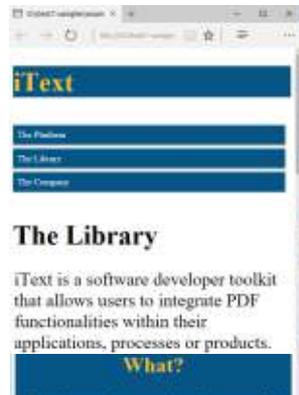


# Example HTML

```
<!DOCTYPE html>
<html>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" type="text/css" href="responsive.css"/>
  <!-- Based on the responsive design example found at https://www.w3schools.com/css/css_rwd_intro.asp -->
</head>
<body>
  <div class="header">
    <h1>iText</h1>
  </div>
  <div class="container">
    <div class="one col-1-m col-1 menu">
      <ul>
        <li>The Platform</li>
        <li>The Library</li>
        <li>The Company</li>
      </ul>
    </div>
    <div class="two col-3-m col-3">
      <h2>The Library</h2>
      <p>iText is a software developer toolkit that allows users to integrate PDF functionalities within their applications, processes or products. </p>
    </div>
    <div class="three col-4-m col-1">
      <div class="aside">
        <h3>What?</h3>
        <p>iText is a software developer toolkit that allows users to integrate PDF functionalities within their applications.</p>
        <h3>How?</h3>
        <p>By providing you with the tools to create and manipulate a pdf in your source code</p>
        <h3>Really?</h3>
        <p>Yes really! </p>
      </div>
    </div>
  </div>
</div>
```

```
<div class="container">
  <div class="header">Some filler text</div>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent ex eros, varius a lorem et, malesuada tempor massa. Amanan elementum rutrum risus vel ultricies. Suspendisse laoreet elit ac purus fringilla porta. Fusce feugiat tortor eget tortor accumsan porta. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; In hac habitasse platea dictumet. Cras tempus mi ullamcorper velit molestie tempus.</p>
  <p>Curabitur pulvinar tincidunt quis maximus tincidunt. Praesent id aliquam tellus. Quisque in ipsum ut tortor sagittis euismod et sit amet dolor. Morrhena vitae tempor felis. Cras convallis justo ac rutrum bibendum. Suspendisse malesuada tincidunt varius. Phasellus non malesuada magna. Phasellus laoreet hendecit magna, sit amet elementum velit fringilla sit amet. Suspendisse risus non, hendecit id lacus eget, tempus ut sem. In hac habitasse platea dictumet. Aliquam erat volutpat. Morrhena laoreet magna a eros sagittis, ac laoreet tortor faucibus.</p>
  <p>Amanan in bibendum justo. Curabitur at mi tempus tincidunt aliquam magna. Praesent venenatis est sit amet mauris egetas accumsan. Proin fringilla leo orci, sit amet euismod justo pellentesque ut. Etiam dignissim nulla id dui porta posuere. Etiam posuere pulvinar suscipit. Donec consectetur massa ac posuere finibus. Cras sit amet sem commodo, molestie eros condimentum, vestibulum justo. Morrhena viverra pulvinar ipsum, non sodales ligula maximus et. Mauris libero leo, ultricies sit amet non, venenatis pellentesque urna. Vestibulum tincidunt mauris ac posuere laoreet. Viverra consectetur alia elit, id euismod magna accumsan non. Praesent tincidunt malesuada tristique. Aliquam sed leo tincidunt tortor facilisis convallis. Donec quis aliquam velit.</p>
  <div class="header">Some small input forms</div>
  <div class="container">
    <div class="col-2-m col-3">
      First name: <input type="text" name="FirstName" value="Lennox"><br>
      Last name: <input type="text" name="LastName" value="PARKER"><br>
    </div>
    <div class="col-2-m col-3">
      Occupation: <input type="text" name="Occupation" value="Paladin"><br>
      Hobbies: <input type="text" name="Hobbies" value="Dragon Hunting"><br>
    </div>
  </div>
</div>
</body>
</html>
```

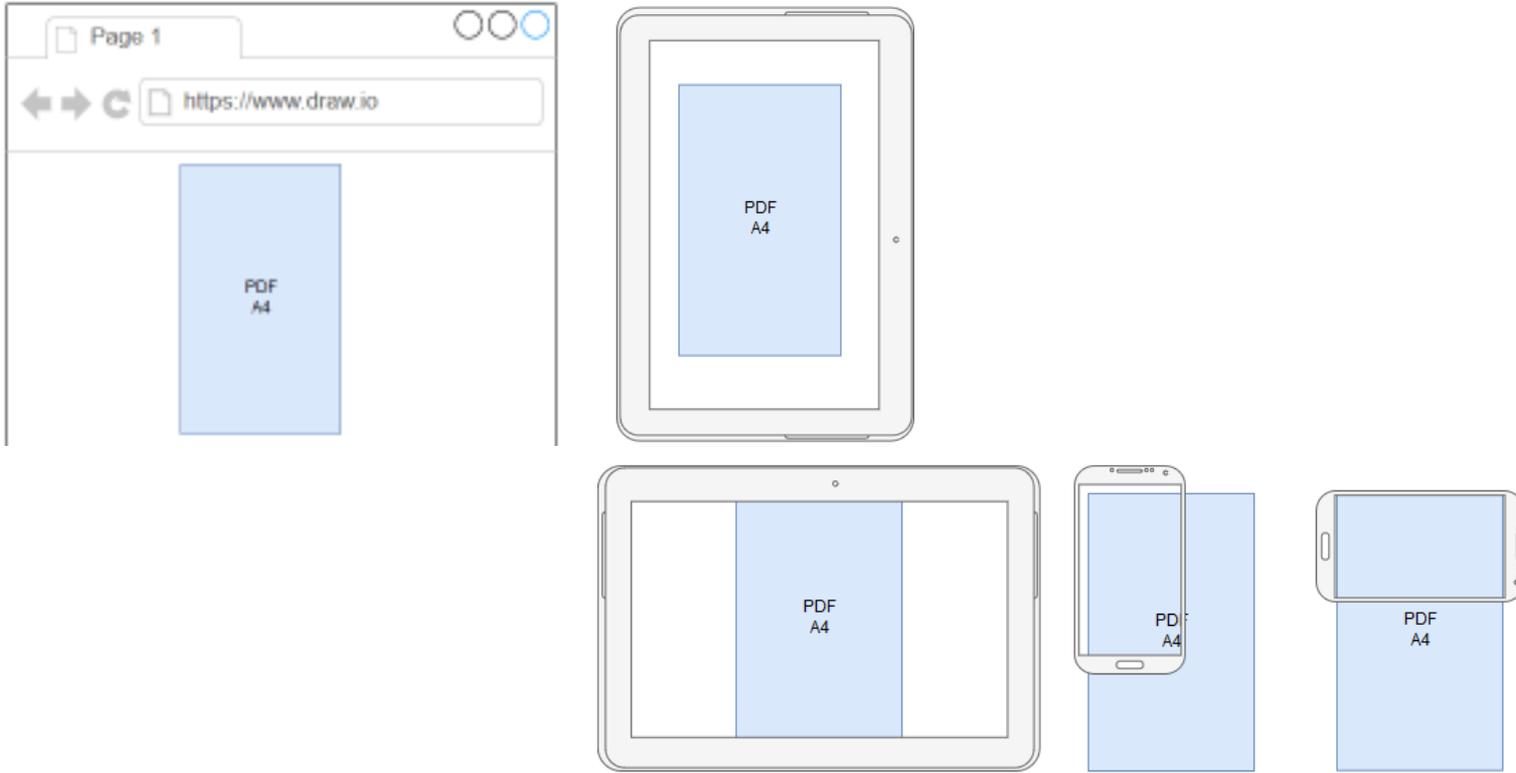
# Example HTML



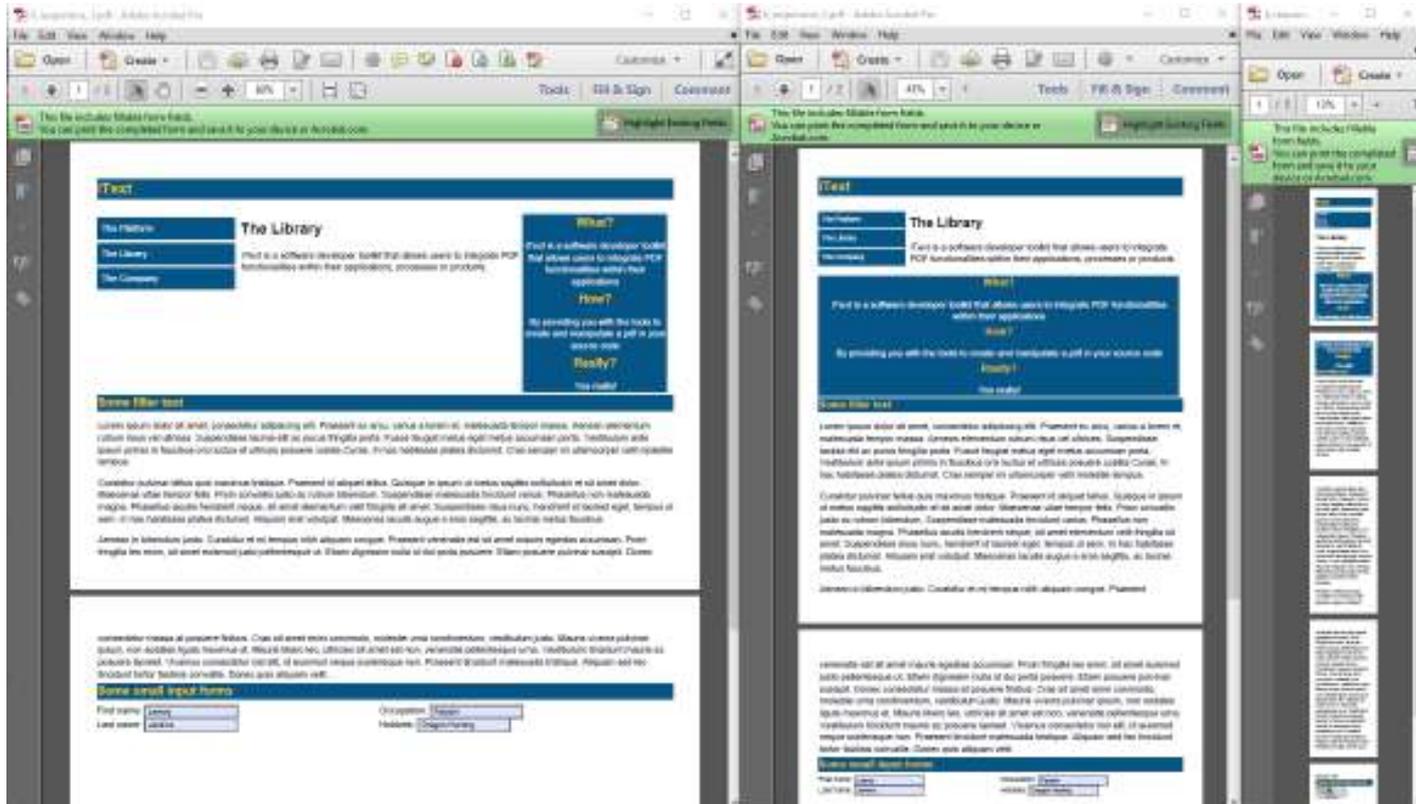
# Media Queries in CSS

```
/*Dynamic sizing*/
[class*="col-"]{
  width: 100%;
}
/*Mobile*/
@media only screen{
  p{
    font-size: 24pt;
  }
  h1{
    font-size: 36pt;
  }
  h2{
    font-size:28pt
  }
}
/*Tablet*/
@media only screen and (min-width: 600px) {
  .col-1-m {width: 24.9%;}
  .col-2-m {width: 49.9%;}
  .col-3-m {width:74%}
  .col-4-m {width: 99%;}
  p{
    font-size: 16pt;
  }
  h1{
    font-size: 24pt;
  }
  h2{
    font-size:18pt
  }
}
/*Desktop*/
@media only screen and (min-width: 768px ) {
  .col-1 {width:24.9%;}
  .col-2 {width: 33.32%;}
  .col-3 {width: 49%;}
  .col-4 {width: 99%;}
  p{
    font-size: 12pt;
  }
  h1{
    font-size: 20pt;
  }
  h2{
    font-size:16pt
  }
}
```

# The problem with PDF on devices



# Convert to different PDFs





# Defining different page sizes

## ☰ Introducing:

### ☰ Tagging

### ☰ Page sizes

### ☰ Media Device Description

### ☰ Create form

```
public static final PageSize[] pageSizes = {
    PageSize.A4.rotate(),
    new PageSize(720, PageSize.A4.getHeight()),
    new PageSize(PageSize.A5.getWidth(), PageSize.A4.getHeight())
};

public static void main(String[] args) throws IOException {
    LicenseKey.loadLicenseFile(System.getenv("ITEXT7_LICENSEKEY") + "/itextkey");
    File file = new File(DEST);
    file.getParentFile().mkdirs();
    E06_MediaQuery app = new E06_MediaQuery();
    int count = 1;
    for (PageSize size : pageSizes) {
        app.createPdf(SRC, String.format(DEST, count++), size);
    }
}

public void createPdf(String src, String dest, PageSize size) throws IOException {
    PdfWriter writer = new PdfWriter(dest);
    PdfDocument pdf = new PdfDocument(writer);
    pdf.setTagged();
    pdf.setDefaultPageSize(size);

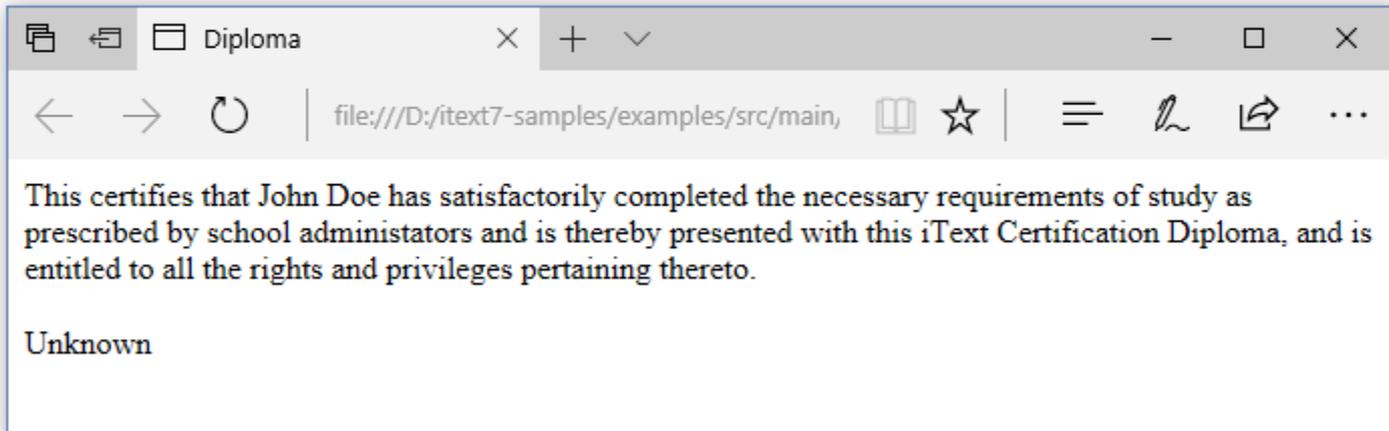
    ConverterProperties properties = new ConverterProperties();
    properties.setBaseUrl(new File(src).getParentFile().getAbsolutePath());
    MediaDeviceDescription mediaDescription = new MediaDeviceDescription(MediaType.SCREEN);
    mediaDescription.setWidth(CssUtils.parseAbsoluteLength(String.valueOf(size.getWidth())));
    properties.setMediaDeviceDescription(mediaDescription);
    properties.setCreateAcroForm(true);
    HtmlConverter.convertToPdf(new FileInputStream(src), pdf, properties);
}
```



# Create your own tags

# Tags student and date don't exist

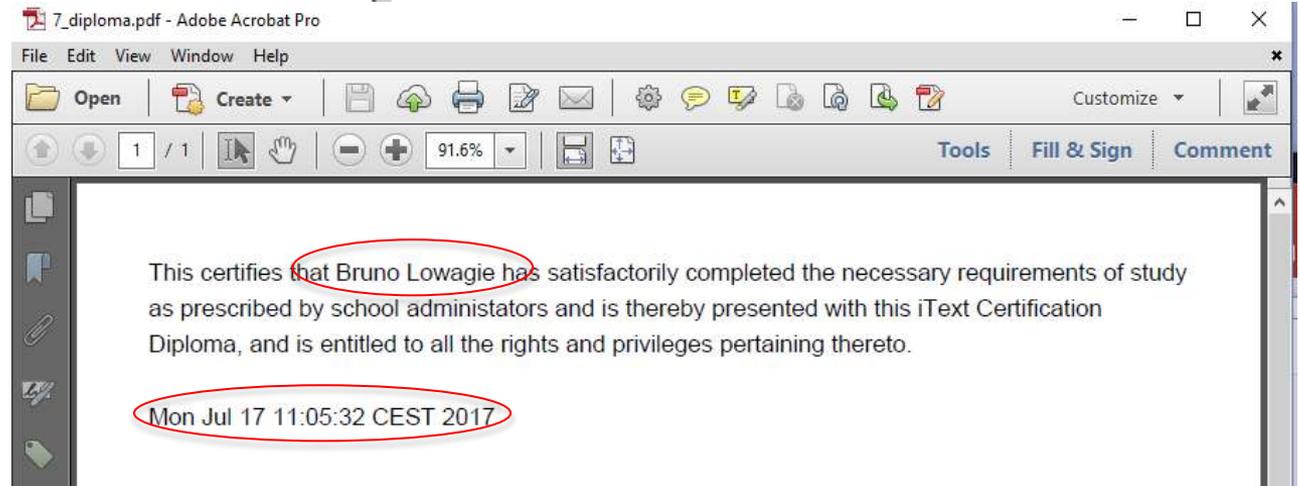
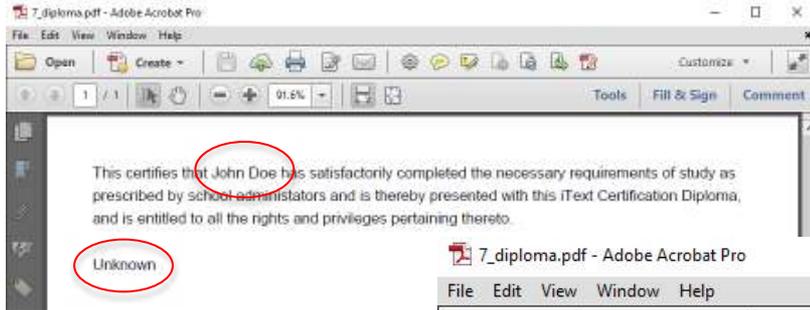
```
<html>
<head><title>Diploma</title></head>
<body>This certifies that <student>John Doe</student> has satisfactorily completed
the necessary requirements of study as prescribed by school administrators and
is thereby presented with this iText Certification Diploma, and is entitled to
all the rights and privileges pertaining thereto.<br><br><date>Unknown</date></body>
</html>
```



# Extend the tag factory

```
public void createPdf(String src, String dest) throws IOException {
    ConverterProperties converterProperties = new ConverterProperties();
    converterProperties.setTagWorkerFactory(
        new DefaultTagWorkerFactory() {
            @Override
            public ITagWorker getCustomTagWorker(IElementNode tag, ProcessorContext context) {
                if ("student".equalsIgnoreCase(tag.name()) ) {
                    return new SpanTagWorker(tag, context) {
                        @Override
                        public boolean processContent(String content, ProcessorContext context) {
                            super.processContent("Bruno Lowagie", context);
                            return true;
                        }
                    };
                }
                else if ("date".equalsIgnoreCase(tag.name()) ) {
                    return new SpanTagWorker(tag, context) {
                        @Override
                        public boolean processContent(String content, ProcessorContext context) {
                            super.processContent(new Date().toString(), context);
                            return true;
                        }
                    };
                }
                return null;
            }
        }
    );
    HTMLConverter.convertToPdf(new FileInputStream(src), new FileOutputStream(dest), converterProperties);
}
```

# Custom tags: custom functionality



# CSS applier needed

```
<terminated> E07_Diploma [Java Application] C:\Program Files\Java\jdk1.8.0_121\bin\javaw.exe (Jul 17, 2017, 2:23:59 PM)
14:23:59.762 [main] ERROR c.i.h.a.impl.DefaultHtmlProcessor - No css applier found for tag student
14:23:59.762 [main] ERROR c.i.h.a.impl.DefaultHtmlProcessor - No css applier found for tag date
```

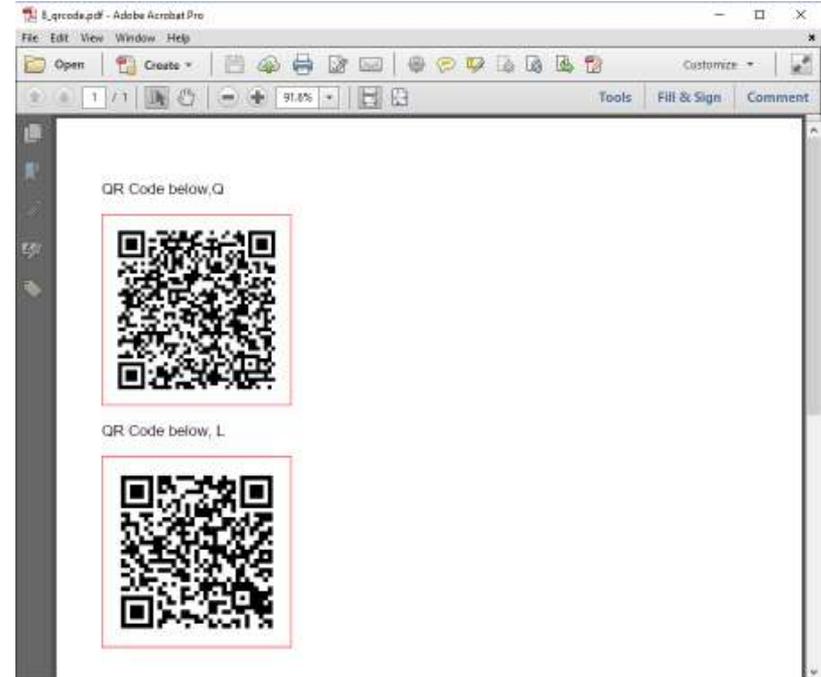
- ☰ More elaborate example: QR Code
  - ☰ QRCodeTagCssApplierFactory extends DefaultCssApplierFactory
  - ☰ QRCodeTagWorkerFactory extends DefaultTagWorkerFactory
  - ☰ QRCodeTagWorker implements ITagWorker

# HTML + CSS => PDF

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>QRCode Example</title>
  <link rel="stylesheet" type="text/css" href="qrcode.css"/>
</head>
<body>
  <p>QR Code below, Q </p>
  <qr charset="Cp437" errorcorrection="Q">
    With great power comes great current squared times resistance
  </qr>

  <p>QR Code below, L</p>
  <qr charset="Cp437" errorcorrection="L">
    With great power comes great current squared times resistance
  </qr>
</body>
</html>

qr{
  border:solid 1px red;
  height:200px;
  width:200px;
}
```



 iText DITO

Under development



# Design Interactive Templates Online

The screenshot displays the iText online design tool interface. The main workspace shows an invoice template for 'iText Group NV'. The invoice includes a header with the company name and address, a large 'Invoice' title, and a table of items. The table has columns for 'Item', 'Quantity', 'Price', and 'Linetotal'. The items listed are Red ball (1, 100), Green ball (2, 100), Orange ball (20, 50), and Purple ball (1, 500). The table also includes a subtotal of 1800, a discount of 60, and a shipping cost of 20, resulting in a total of 1760. The interface also shows a left sidebar with a project list, a top navigation bar, and a right sidebar with a properties panel and a data panel.

**iText Group NV**  
Business Center "De Punt"  
Kerkstraat 105  
9050 Gentbrugge (Ghent)  
Belgium  
<http://itextpd.com/sales@itextpd.com>  
+32 9 298 02 31

Due date: 2018-01-30

To client  
{invoiceDate}  
{customer.name}  
{customer.address.postalCode}

Serial payment to  
Account: 236700  
IBAN: NL2548N40947340030  
BIC: ICSG3333

## Invoice

2015-12-01

Item	Quantity	Price	Linetotal
Red ball	1	100	100
Green ball	2	100	200
Orange ball	20	50	1000
Purple ball	1	500	500
SUBTOTAL			1800
DISCOUNT			60
SHIPPING			20
<b>TOTAL</b>			<b>1760</b>

Properties panel:  
Name: Invoice2  
Location: /  
Data Source Template: Invoice2  
Output Language: [empty]  
Output Type: web-html5  
Data panel:  
Filter: [empty]  
invoiceDate  
identifier  
paymentDueDate  
accountId

# Design Interactive Templates Online

The screenshot displays the iText online design tool interface. The main workspace shows a preview of an invoice template. The invoice is for iText Group NV, with a due date of 2018-01-30 and an invoice date of 2015-12-01. The invoice includes a table of items: Red ball (1 unit, 100 price, 100 net total) and Green ball (2 units, 100 price, 200 net total). The total amount is 260. The interface also shows a sidebar with a list of pages (Text 2, Order table) and a right-hand panel with data source and output settings.

**iText Group NV**  
Business Center "De Punt"  
Kerkstraat 108  
9050 Gentbrugge (Ghent)  
Belgium  
<http://itextpdf.com/>  
[sales@itextpdf.com](mailto:sales@itextpdf.com)  
+32 9 298 02 31

Due date: 2018-01-30

To client  
{invoiceDate}  
{customer.name}  
{customer.address.postalCode}

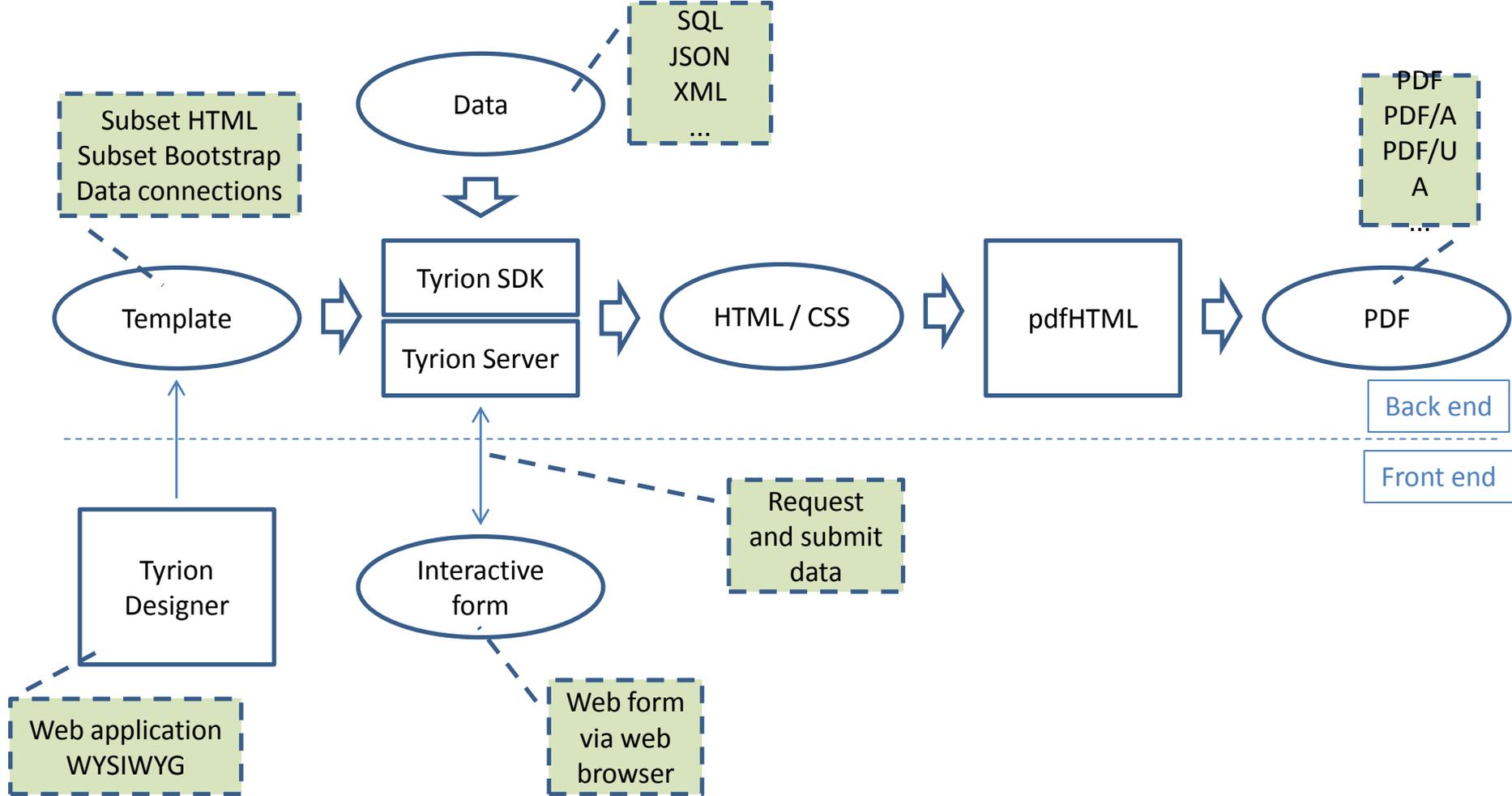
Serial payment to  
Account: 256799  
IBAN: NL25ABNA0847346530  
BIC: GEBFN333

## Invoice

2015-12-01

Item	Quantity	Price	Linetotal
Red ball	1	100	100
Green ball	2	100	200
<b>SUBTOTAL</b>			300
<b>DISCOUNT</b>			60
<b>SHIPPING</b>			20
<b>TOTAL</b>			<b>260</b>

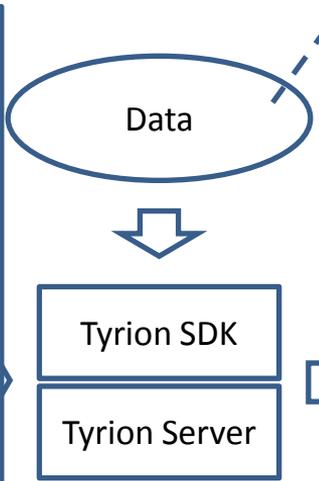
thank you very much. We really appreciate your business  
Please send payment within 21 days of receiving this invoice



**iTEXT**

<Name>  
<Address>

<Invoice items table  
(4 cols, headers, ...)>  
<Total>



**iTEXT**

Raf Hens  
Kerkstraat 108  
9050 Gentbrugge

item	qty	price	tot
Item A	4	100	400
Item B	1	10	10
Item C	17	45	765
Item D	2	50	100
Item E	1	70	70
Item F	4	250	1000
Item G	5	100	500
Item H	12	3	36
Item I	1	100	100
Item J	1	35	35
Item K	1	250	250

**3266**



**iTEXT**

Raf Hens  
Kerkstraat 108  
9050 Gentbrugge

item	qty	price	tot
Item A	4	100	400
Item B	1	10	10
Item C	17	45	765
Item D	2	50	100
Item E	1	70	70
Item F	4	250	1000
Item G	5	100	500

p 1/2



Web application  
WYSIWYG

**iTEXT**

<Name>  
<Address>

item	qty	price	tot
<Item>	<Qty>	<Price>	<Tot>
<Item>	<Qty>	<Price>	<Tot>

<Total>

**iTEXT**

item	qty	price	tot
Item H	12	3	36
Item I	1	100	100
Item J	1	35	35
Item K	1	250	250

**3266**

p 2/2



Thank you!