



# Support of complex scripts in PDF

Lego game of text composition and text  
extraction algorithms

Alexey Subach  
iText Group NV



မင်္ဂလာညနေခင်းပါ

مساء الخير

Добрый вечер

सुसंध्या

ערב טוב

Good evening

สวัสดีตอนเย็น

Labas vakaras

God kväll

អរុណសួស្តី

शुभ सक्ता

Buonasera

# Complex scripts *are* complex!

To get this:

تَعْلِيمُ اللُّغَةِ الْعَرَبِيَّةِ لَغَيْرِ النَّاطِقِينَ بِهَا

Instead of that:

أهـبـ نـيـقـطـأـنـلـاـرـيـغـلـةـيـبـرـعـلـاـةـغـلـلـاـمـيـلـعـت

...you have got a long way to go

# Steps in processing complex scripts

- ≡ Cluster characters into syllables or words
- ≡ Apply Unicode character reordering
- ≡ Apply Open Type features (shaping)
- ≡ Apply *layout-related* processing, e.g. hyphenation
- ≡ Apply Unicode Bidi algorithm
- ≡ Write content using correct syntax (to view and extract data)
- ≡ Create tag structure
- ≡ Extract content back from PDF to Unicode

# Steps in processing complex scripts

- ≡ Cluster characters into syllables or words
- ≡ Apply Unicode character reordering
- ≡ **Apply Open Type features (shaping)**
- ≡ Apply *layout-related* processing, e.g. hyphenation
- ≡ **Apply Unicode Bidi algorithm**
- ≡ **Write content using correct syntax (to view and extract data)**
- ≡ Create tag structure
- ≡ **Extract content back from PDF to Unicode**

# Example: Arabic text

أهـب نـيـق طـأ نـل اـري غـل ةـي بـرـعـل اـة غـل لـاـمـي لـعـت  
تـعـلـيـمُ الـلـغـة الـعـرـبـيـة لـغـيـر الـنـاـطـقـيـن بـةـا  
تـعـلـيـمُ الـلـغـة العـرـبـيـة لـغـيـر النـاـطـقـيـن بـهـا  
تـعـلـيـمُ الـلـغـة العـرـبـيـة لـغـيـر النـاـطـقـيـن بـهـا

# Example: Khmer text

ហាដើហ៊ានចនាសម័នធិត្ថវកាវ  
ហោដ្ឋានចនាសម័នត្រ័វកាវ  
ហោដ្ឋានចនាសម័នត្រ័វកាវ

# Bidi algorithm

Logical representation

סלוע סולש



Visual representation

שולום עולם



Unicode Bidirectional Algorithm



# Bidi algorithm

≡ Mixed directions in text: numbers and English text are still shown left-to-right

October PDFest 2020 שלום



# Bidi algorithm

≡ Mixed directions in text: numbers and English text are still shown left-to-right

October PDFest 2020 שלום



≡ **/ReversedChars:** informs the PDF processor that *show strings* within a marked-content sequence *contain characters in reverse order*. In order for such text to be extracted or read out aloud, the sequence of the characters as found in the show string operator shall be reversed before using them

# Bidi algorithm

≡ Character mirroring

≡ Unicode:

`\u05E9...\u05DD` (`\u05DB\u05DC ... \u05EA`)

שלום עולם) כל  
(היבשות



≡ /ToUnicode – does not work if there are both mirrored and regular brackets

≡ Using multiple fonts – bugs when editing the PDF (e.g. form fields)

# Bidi algorithm

≡ Character mirroring

≡ Unicode:

`\u05E9...\u05DD` (`\u05DB\u05DC ... \u05EA`)

שלום עולם) כל  
(היבשות

≡ /ToUnicode – does not work if there are both mirrored and regular brackets

≡ Using multiple fonts – bugs when editing the PDF (e.g. form fields)

≡ /ActualText

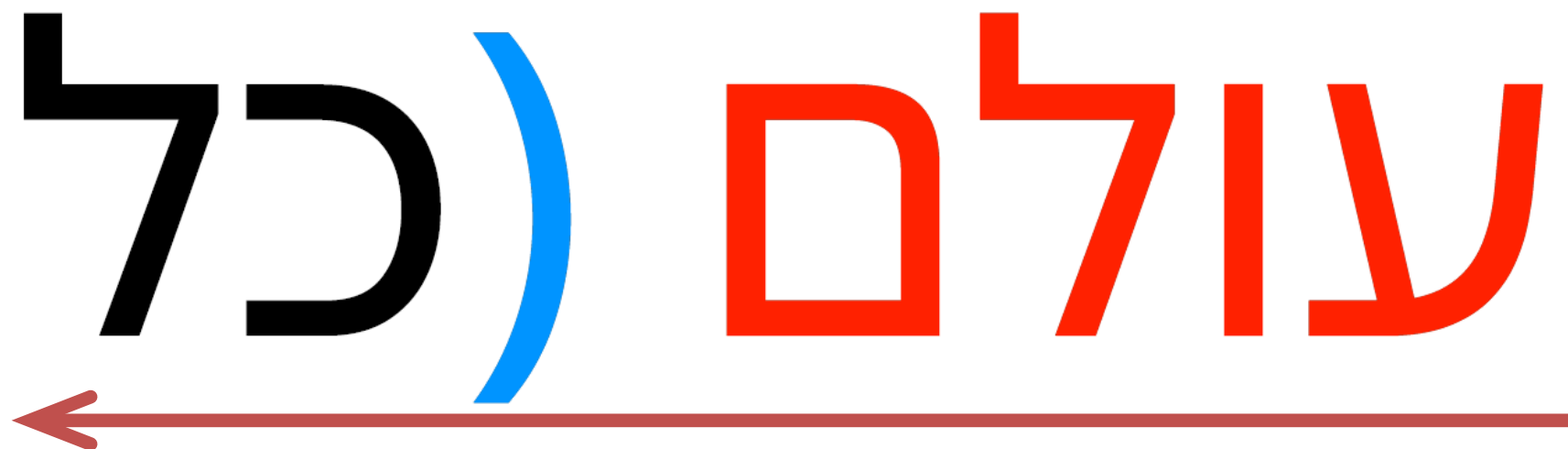
# Is /ActualText our savior?

- ≡ Can be specified for content that does translate into text but that is represented in a nonstandard way (*ISO 32000-2*)
- ≡ Replacement text can be specified for the following items:
  - ≡ A structure element, by means of the optional **ActualText** entry (*PDF 1.4*) of the structure element dictionary.
  - ≡ (*PDF 1.5*) A marked-content sequence, through an **ActualText** entry in a property list attached to the marked-content sequence with a **Span** tag.

# Example

```
BT
/F2 30 Tf
91.75 700.03 Td
/Span <</ActualText <feff0028> >> BDC # Mirroring the bracket for extraction
<000c> Tj # ")" is shown, "(" is extracted
EMC
ET
```

```
BT
/F1 30 Tf
108.75 700.03 Td
/ReversedChars BMC
<001b001a00130020> Tj
EMC
ET
```



# RTL Text Extraction: In practice

- ≡ If you don't use `/ReversedChars`: **many major** PDF viewers/consumers will still understand that this is RTL word
- ≡ If you use `/ReversedChars`: behavior of **many** PDF viewers/consumers is exactly the same as if you did not
- ≡ If you don't use `/ActualText` to mirror the paired characters: **some** PDF consumers will still apply the inverse Bidi algorithm and extract the correct logical sequence
- ≡ If you use `/ReversedChars` with `/ActualText` for mirroring, **very few** PDF consumers will understand the intention when performing text extraction and typically brackets will be mirrored even though they were not intended to

# RTL text: correct tagging

≡ *Content shall be marked in the structure tree with semantically appropriate tags in a logical reading order (ISO 14289)*

- Document

- P

- **Span**

- **Span**

October PDFest 2020 שלום





# Inverse Bidi algorithm

- ≡ Restores logical sequence from visual one
- ≡ Needed in case **/ReversedChars** is missing
- ≡ Not necessarily whole text chunk in **Tj**, glyphs can be in random order which requires sorting in the first place
- ≡ Even for non-random order you need to detect text orientation for independent glyphs if they are in separate text showing operators
- ≡ Many-to-one function, results may be unexpected

# Inverse Bidi algorithm

Sort the glyphs by their position and feed them to the algorithm!

October PDFest 2020 שלום

From **visual**

**1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24**

to **logical**

**6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 5 4 3 2 1**

# On the way to success

أهـبـ نـيـقـ طـأـنـلـ اـرـيـ غـلـ ةـيـ بـرـعـلـ اـةـ غـلـلـ اـمـيـلـ عـتـ



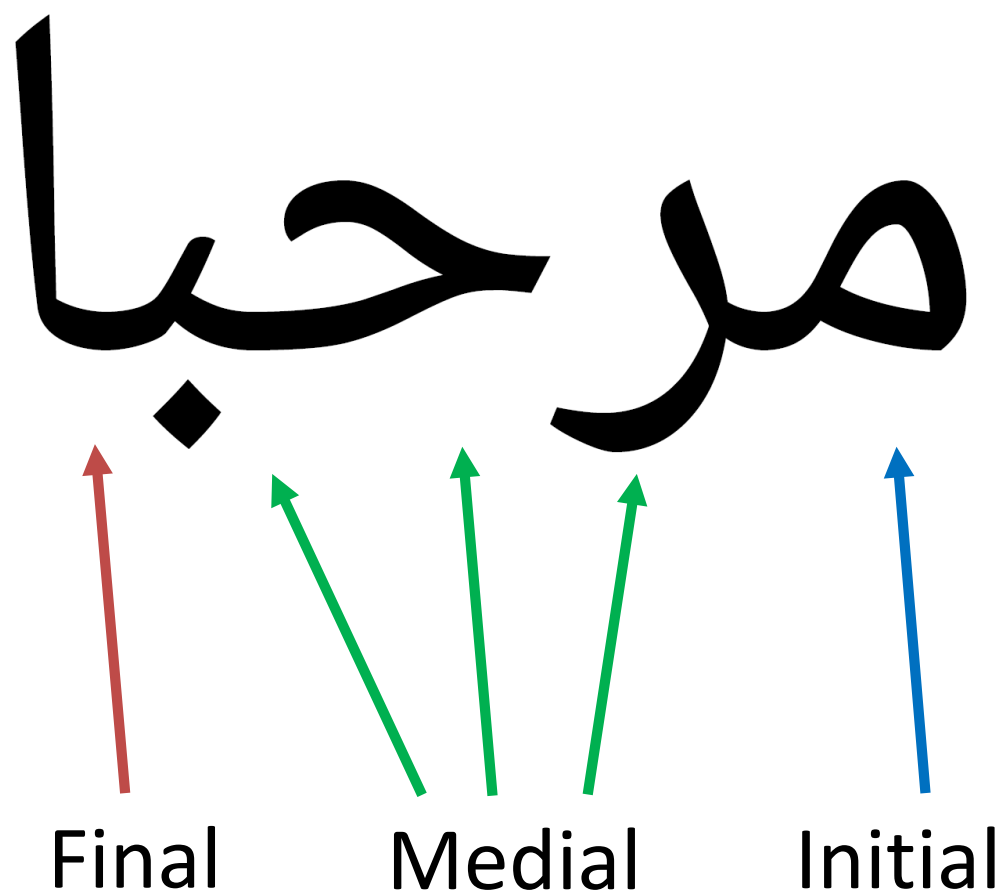
تـعـلـيـمـ اـلـلـغـةـ اـلـعـرـبـيـةـ لـغـيـرـ اـلـنـاـطـقـيـنـ بـهـاـ



تـعـلـيـمـ اللـغـة العـرـبـيـة لـغـيـر النـاـطـقـيـن بـهـاـ

تـعـلـيـمـ اللـغـة العـرـبـيـة لـغـيـر النـاـطـقـيـن بـهـاـ

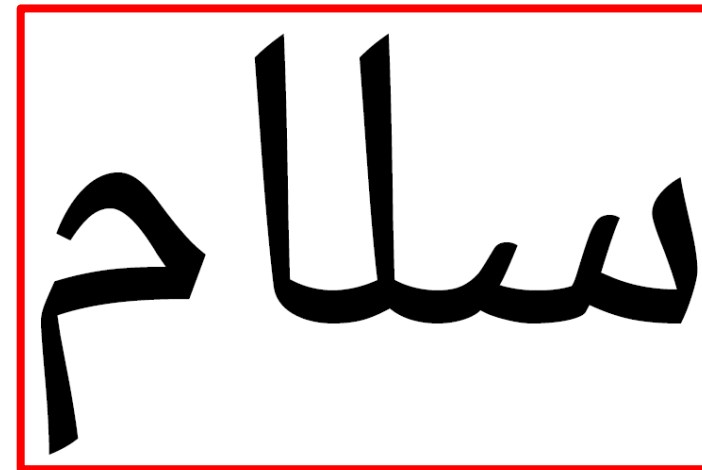
# Arabic shaping: 4 glyph forms



Isolated	Final	Medial	Initial
ﺀ	ﺀ	ﺀ	ﺀ
ﺐ	ﺐ	ﺐ	ﺐ
ﺒ	ﺒ	ﺒ	ﺒ
ﺒـ	ﺒـ	ﺒـ	ﺒـ
ﺒ	ﺒ	ﺒ	ﺒ

# OpenType Implementation

- ≡ OpenType fonts contain tables with the relevant information: **init**, **medi**, **fini** features in **GSUB** table
- ≡ Fonts must be of good quality as they provide vital information for shaping
- ≡ Do not forget about **rlig** (*Required Ligatures*) feature





# What if fonts are lacking some info?

- ≡ Some operations can be done purely on Unicode level, e.g. Arabic shaping
- ≡ Extensive Unicode information is a requirement in that case (1442 code points)
  - ≡ Arabic Presentation Forms-A (FB50–FDFF, 611 characters)
  - ≡ Arabic Presentation Forms-B (FE70–FEFF, 141 characters)
- ≡ Default (isolated), initial, medial, final, forms of glyphs require 4x characters in Unicode

# What if fonts are lacking some info?

- ≡ Some operations can be done purely on Unicode level, e.g. Arabic shaping
- ≡ Extensive Unicode information is a requirement in that case (1442 code points)
  - ≡ Arabic Presentation Forms-A (FB50–FDFF, 611 characters)
  - ≡ Arabic Presentation Forms-B (FE70–FEFF, 141 characters)
- ≡ Default (isolated), initial, medial, final, forms of glyphs require 4x characters in Unicode
- ≡ Arabic Letter Khah Medial Form (U+FEA8) ﺍ̤

# What if fonts are lacking some info?

- Some operations can be done purely on Unicode level, e.g. Arabic shaping
- Extensive Unicode information is a requirement in that case (1442 code points)
  - Arabic Presentation Forms-A (FB50–FDFF, 611 characters)
  - Arabic Presentation Forms-B (FE70–FEFF, 141 characters)
- Default (isolated), initial, medial, final, forms of glyphs require 4x characters in Unicode
- Arabic Letter Khah Medial Form (U+FEA8) 
- Arabic Ligature Salam Isolated Form (U+FD55) 



# What if fonts are lacking some info?

- Some operations can be done purely on Unicode level, e.g. Arabic shaping
- Extensive Unicode information is a requirement in that case (1442 code points)
  - Arabic Presentation Forms-A (FB50–FDFF, 611 characters)
  - Arabic Presentation Forms-B (FE70–FEFF, 141 characters)
- Default (isolated), initial, medial, final, forms of glyphs require 4x characters in Unicode
- Arabic Letter Khah Medial Form (U+FEA8) خ
- Arabic Ligature Salam Isolated Form (U+FD55) صل
- Arabic Ligature Bismillah Ar-Rahman Ar-Rageem (U+FD5D) بسم الله الرحمن الرحيم

# Not for all complex scripts

## ≡ Substitution types

- ≡ One to one (Arabic shaping)
- ≡ Many to one (ligatures)
- ≡ Many to many (incl. context-aware)
- ≡ One to many (decomposition)

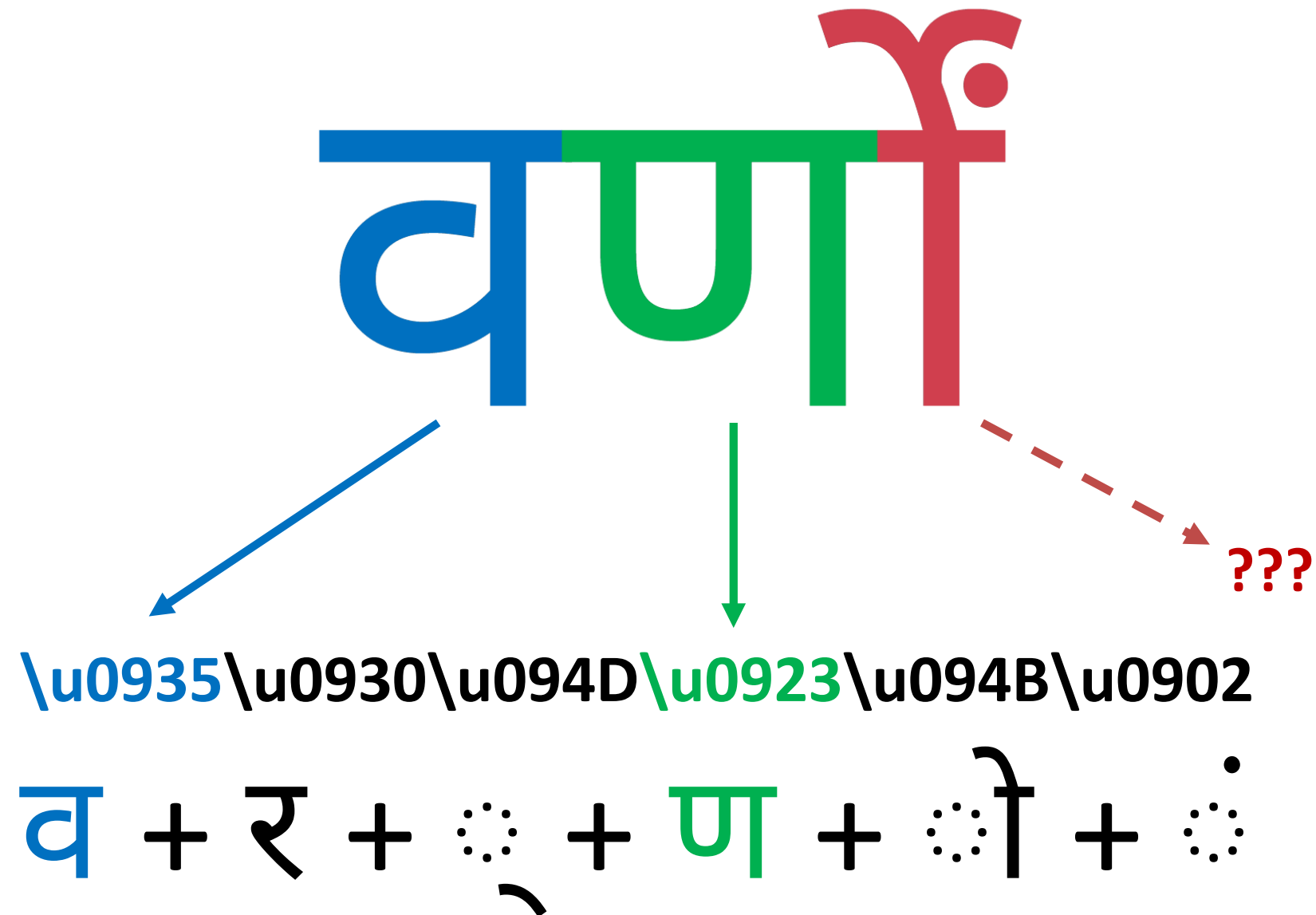
≡ 691 substitutions for Arabic in a font compared to 1270 substitutions for a Devanagari font while Devanagari only has 208 Unicode characters reserved

≡ Impractical to store all the combinations at the Unicode level, contextual substitutions make it even harder, font lookups are more flexible

# PDF Implementation

- ≡ Easy for initial, medial, final forms – just use **/ToUnicode** to map to the default form
- ≡ No Unicode mapping for some of the glyphs from more complex substitutions (e.g. Devanagari)
- ≡ Reordering at Unicode level for clusters before applying font features makes ligature-like **/ToUnicode** impossible
- ≡ Syllable (cluster) is a minimal unit that can guarantee correspondence between visual representation and Unicode sequence that you get on copy-paste

# Text extraction: getting trickier



# /ActualText is our savior!

- ≡ Turns out in many cases it **has** to be specified
- ≡ You should use the minimal possible sequence in the /ActualText block to allow copying small chunks of text

```
/Span <</ActualText <FEFF 0930 094D 0923 094B 0902> >> BDC  
<002701C4>Tj  
EMC
```

# /ActualText: In practice

- ≡ Not supported in many PDF viewers
- ≡ Problems with determining spaces when extracting text in some viewers that support /ActualText
- ≡ Problems with extracting diacritical marks when extracting text in some viewers that support /ActualText
- ≡ If each of two (or more) consecutive structure or marked-content sequences has an **ActualText** entry, they shall be treated as if no word break is present between them (**ISO 32000-2**)

# On the way to success

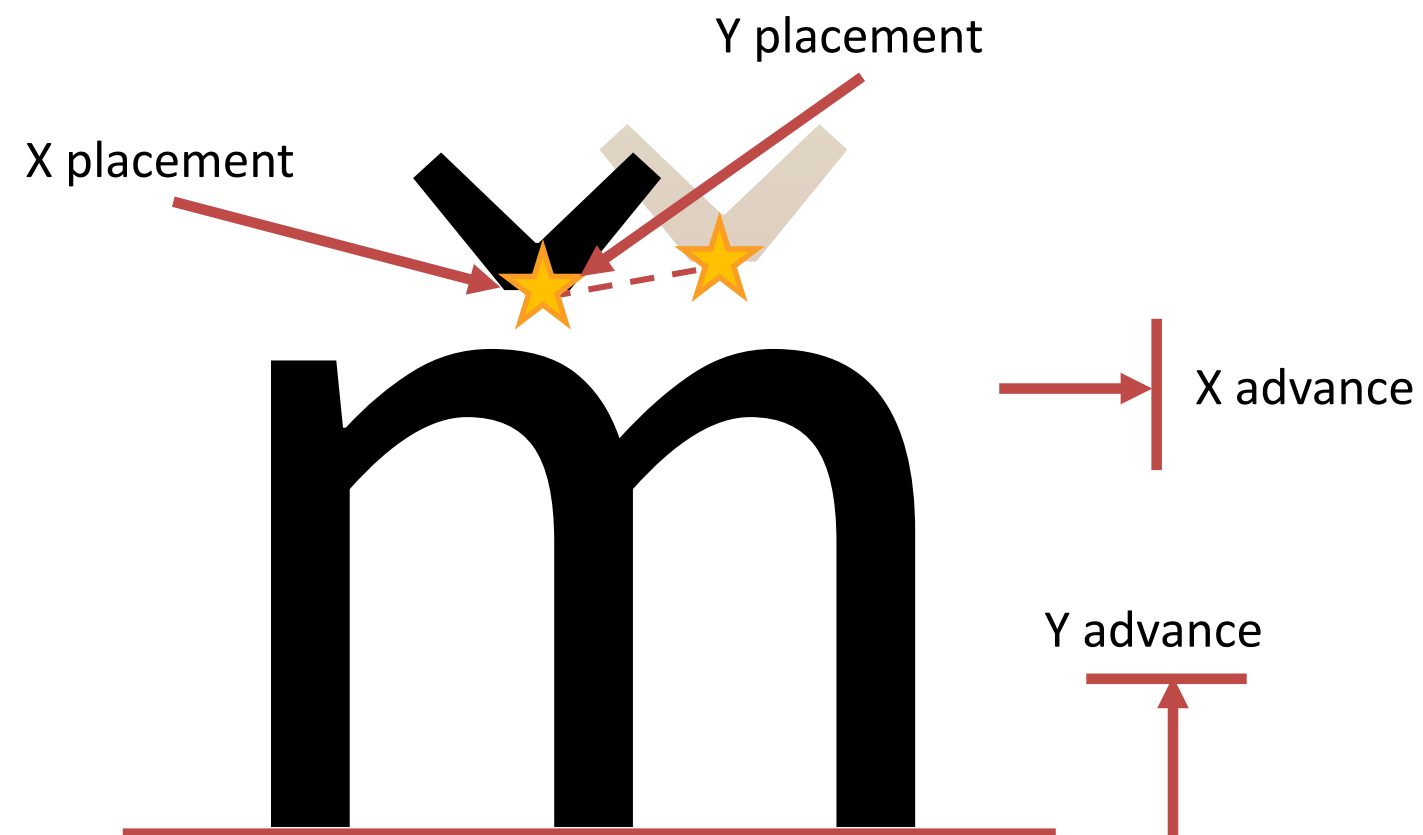
أهـبـ نـيـقـ طـأـنـلـ اـريـ غـلـ ةـيـ بـرـعـلـ اـةـ غـلـلـ اـمـيـلـ عـتـ

تـعـلـيـمـ اـلـلـغـةـ اـلـعـرـبـيـةـ لـغـيـرـ اـلـنـاـطـقـيـنـ بـهـاـ

تـعـلـيـمـ اللـغـةـ العـرـبـيـةـ لـغـيـرـ النـاـطـقـيـنـ بـهـاـ

تـعـلـيـمـ اللـغـةـ العـرـبـيـةـ لـغـيـرـ النـاـطـقـيـنـ بـهـاـ

# Attaching glyph to one another



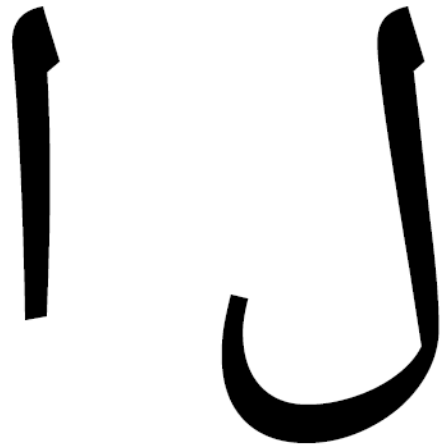


# Mark to ligature attachment

- ≡ Several possible attachment components force us into tracking which ligature component the mark is associated with
- ≡ /ActualText plays vital role in extracting the text correctly

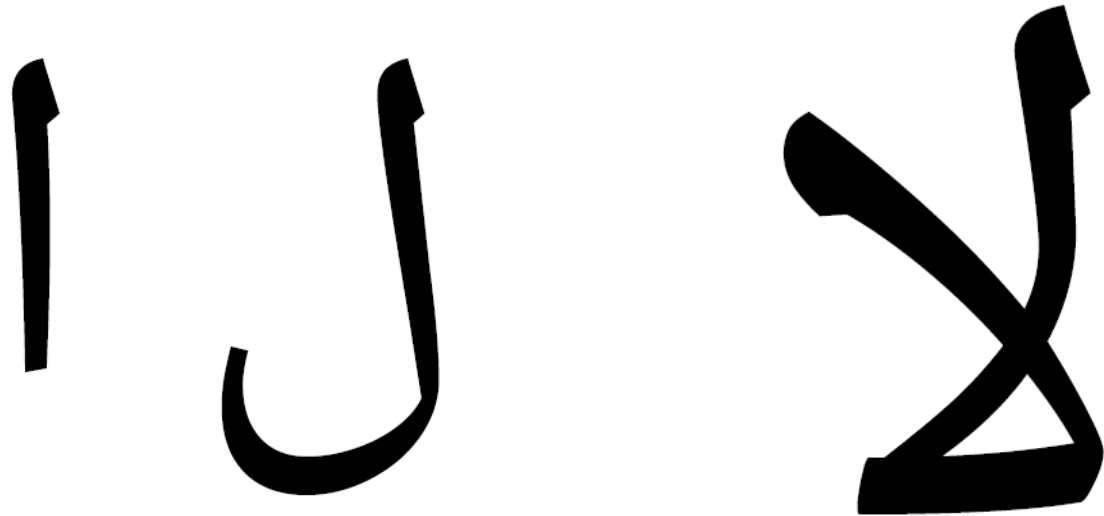
# Mark to ligature attachment

- ≡ Several possible attachment components force us into tracking which ligature component the mark is associated with
- ≡ /ActualText plays vital role in extracting the text correctly



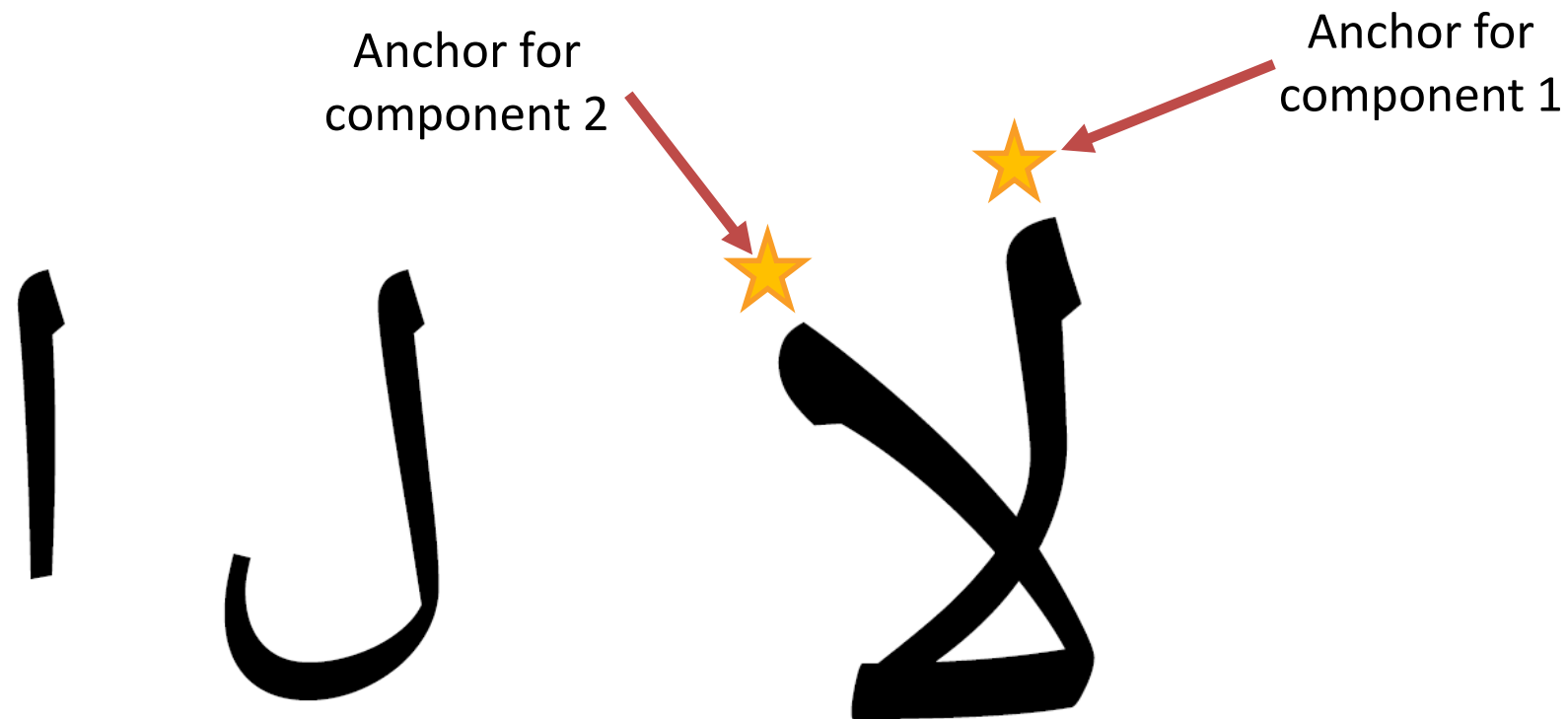
# Mark to ligature attachment

- ≡ Several possible attachment components force us into tracking which ligature component the mark is associated with
- ≡ /ActualText plays vital role in extracting the text correctly



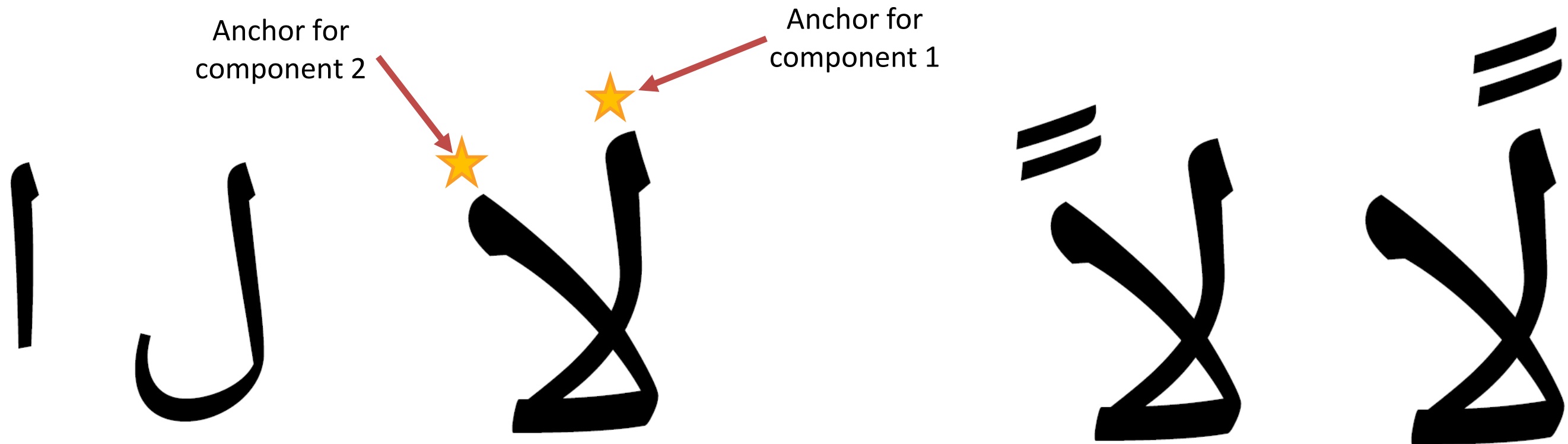
# Mark to ligature attachment

- ≡ Several possible attachment components force us into tracking which ligature component the mark is associated with
- ≡ /ActualText plays vital role in extracting the text correctly



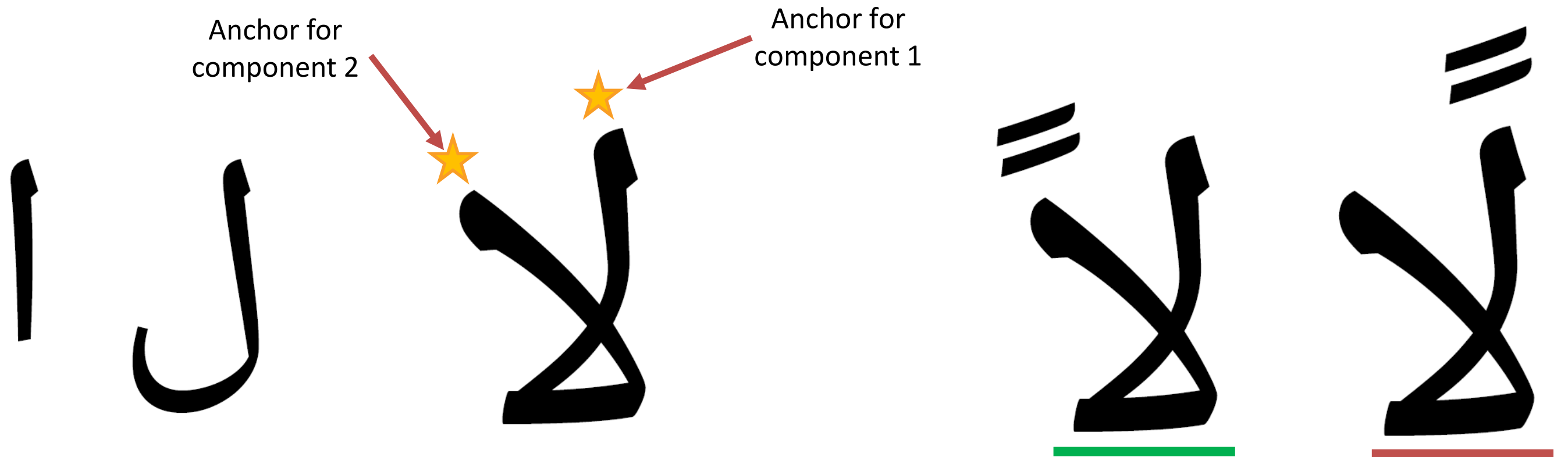
# Mark to ligature attachment

- ≡ Several possible attachment components force us into tracking which ligature component the mark is associated with
- ≡ /ActualText plays vital role in extracting the text correctly



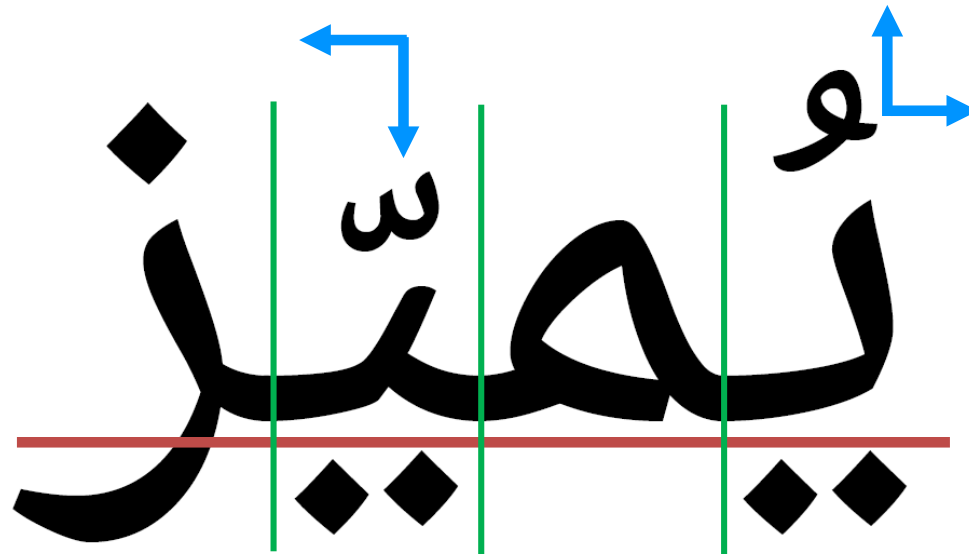
# Mark to ligature attachment

- ≡ Several possible attachment components force us into tracking which ligature component the mark is associated with
- ≡ /ActualText plays vital role in extracting the text correctly



# Extracting base glyphs + marks

- It is a common practice to perform some glyph sorting as a part of text extraction pipeline
- Glyph positioning: offsets and attachments affect the sorted order of the glyphs, both along X and Y axes, both mark to base and mark to mark
- Sorting needs to be adjusted to take that into account



# Conclusions

- ≡ All the pieces of the puzzle must fall together for the complex scripts to be supported (assuming the font is of good quality already 😊 )
- ≡ Poorly-generated documents provide a very limited possibility for extracting data from them in automatic way
- ≡ Not supporting complex scripts in products we make creates interoperability problems and might slow down adoption of PDF as the *universal* document interchange format in countries natively using complex scripts



# References

- ≡ PDF and OpenType, PDF Days 2016 talk - <https://www.youtube.com/watch?v=qZnXZppH2KI>
- ≡ OpenType specification - <https://www.microsoft.com/en-us/Typography/OpenTypeSpecification.aspx>
- ≡ Microsoft Typography - <https://www.microsoft.com/en-us/Typography/default.aspx>
- ≡ FontForge Open Source tool - <https://fontforge.org>
- ≡ OpenType CookBook - <http://opentypecookbook.com>



Solutions offered by iText

# Who we are

**20 YEARS**  
of iText Code  
In 2020

- ≡ Leading PDF technology company
- ≡ Board member of PDF Association
- ≡ Open Source SDK is available for 20 years
- ≡ 125M users worldwide
- ≡ 7000+ clients worldwide



#### **iText 7 Core**

Process PDF documents in Java or .NET



#### **iText 7 add-on: pdfXFA**

Flatten dynamic XFA forms to static PDF



#### **iText 7 add-on: pdf2Data**

Extract data from templated PDF documents



#### **iText 7 add-on: pdfHTML**

Convert HTML and CSS into PDF documents



#### **iText 7 add-on: pdfCalligraph**

Unlock advanced features in PDF typography



#### **iText 7 add-on: pdfRender**

Generate images from PDF



#### **iText 7 add-on: pdfSweep**

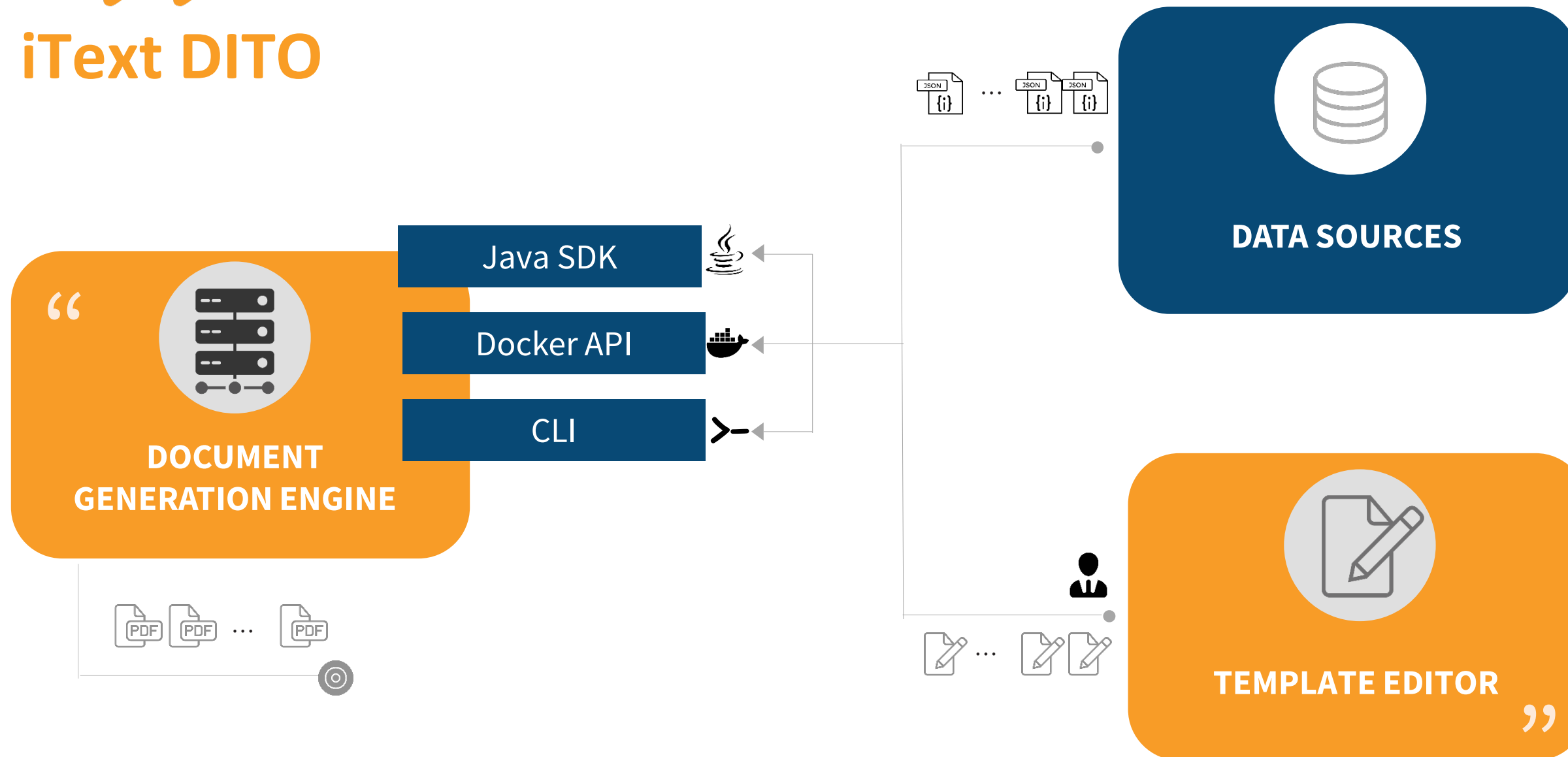
Securely redact data from PDF documents



#### **iText 7 add-on: pdfOCR**

Convert images into searchable PDF documents

# ” iText DITO



**Object 6 0 R -> Dictionary of type: /Page**

- Page 1
  - /Contents: 7 0 R -> Stream
    - Stream**
      - /Filter: /FlateDecode
      - /Length: 3358
  - /MediaBox: Array
  - /Parent: 2 0 R -> Dictionary of type: /Pages
  - /Resources: Dictionary

**Structure Tree**

- /Document
  - /P
    - /Span

Key	Value	
/Filter	/FlateDecode	✗
/Length	3358	✗
/		+

Stream
XFA
Debug info
Console

```

<0040> Tj
2.676 0 Td
<0003000e002e003400030034004b001a00420003002c> Tj
54.684002 0 Td
/Span <</ActualText <feff093909400902> >> BDC
<003d01ad> Tj
EMC
9.468 0 Td
<0003003d004f002800440003003d004c02e40003000b003c004b0003> Tj
56.160004 0 Td
/Span <</ActualText <feff0936093f> >> BDC
<0233003a> Tj
EMC
11.256 0 Td
<0034004f0034004b0040001a004200030019003d0028004b0003> Tj
54.372002 0 Td
/Span <</ActualText <feff093909480902> >> BDC
<003d01b9> Tj
EMC
6.372 0 Td
<00680003000b003c001900420003> Tj
36.720001 0 Td
/Span <</ActualText <feff0935093f> >> BDC
<02310039> Tj
EMC

```

\* Screenshot from iText RUPS <https://github.com/itext/i7j-rups>

```
PdfDocument pdfDocument = new PdfDocument(new PdfWriter(PDF_PATH));
Document document = new Document(pdfDocument);

document.add(new Paragraph("Good evening to OctoberPDFest participants!"));

Table greetingsTable = new Table(2)
    .setBackgroundColor(ColorConstants.LIGHT_GRAY);
GREETINGS.forEach((language, text) ->
    greetingsTable.addCell(language).addCell(text));

document.add(greetingsTable.setHorizontalAlignment(HorizontalAlignment.CENTER));

document.close();
```

---

Good evening to OctoberPDFest participants!

French	Bonsoir
Italian	Buonasera
German	Guten Abend
Lithuanian	Labas vakaras



```
Collection<String> FONTS = Arrays.asList(ARABIC_FONT_PATH,  
    DEVANAGARI_FONT_PATH, THAI_FONT_PATH, KHMER_FONT_PATH);  
FontProvider fontProvider = new FontProvider();  
FONTS.forEach(fontProvider::addFont);  
document.setFontFamily("Noto").setFontProvider(fontProvider);
```

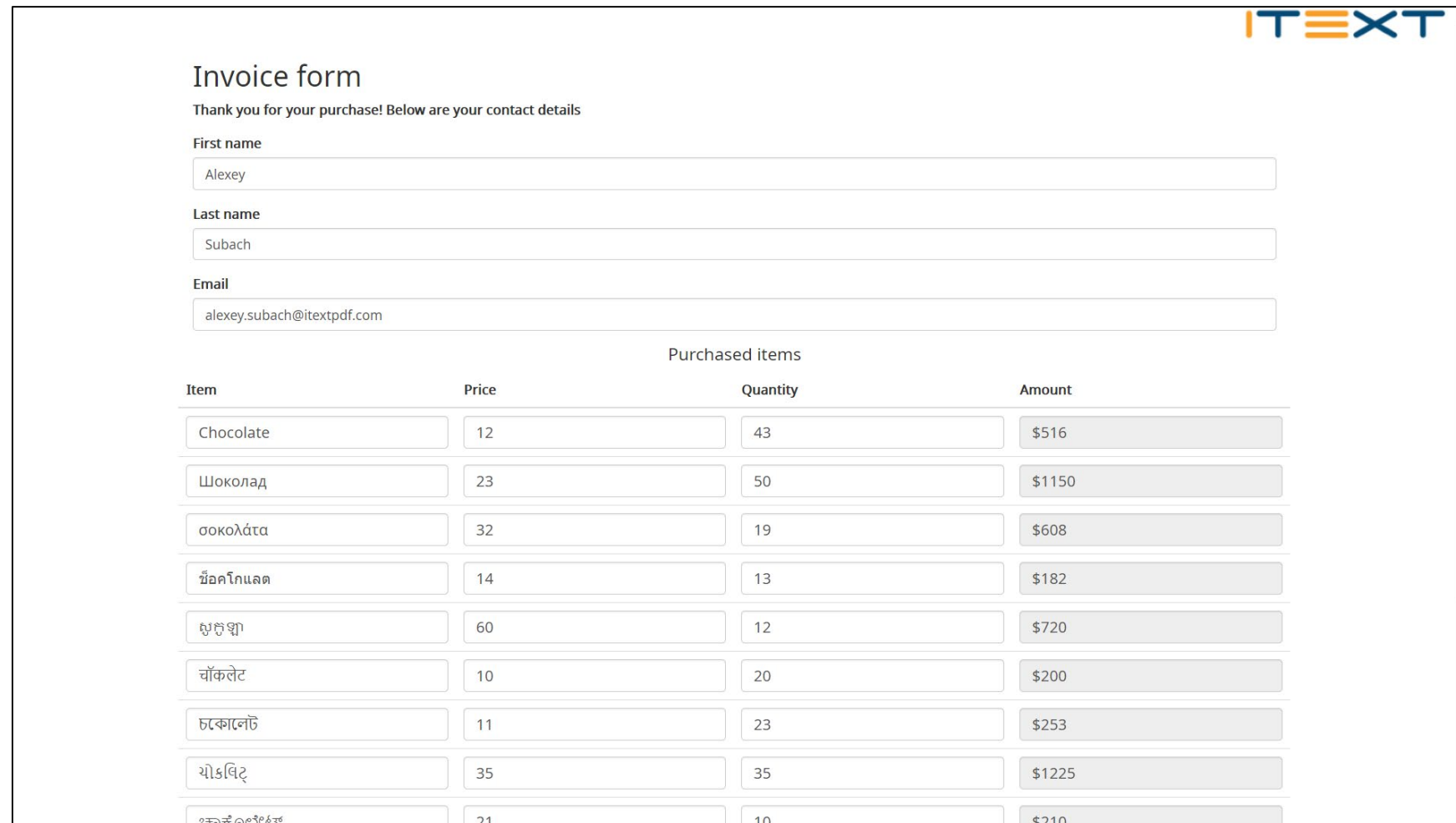
```
FontProvider fontProvider = new FontProvider();  
fontProvider.addSystemFonts();  
document.setFontFamily("Noto Sans").setFontProvider(fontProvider);
```

Good evening to OctoberPDFest participants!

Devanagari	सुसंध्या
Thai	สวัสดีตอนเย็น
Khmer	អរុណសួស្តី
Arabic	مساء الخير

# pdfHTML

- ≡ HTML as more high-level syntax
- ≡ Invoices, reports, contracts and other data from CRM, ERP systems



The screenshot displays a web form for generating an invoice. It includes input fields for 'First name' (Alexey), 'Last name' (Subach), and 'Email' (alexey.subach@itextpdf.com). Below these is a table titled 'Purchased items' with columns for 'Item', 'Price', 'Quantity', and 'Amount'. The table lists various chocolate items in multiple languages, including English, Russian, Greek, Thai, Bengali, Hindi, and Chinese.

Item	Price	Quantity	Amount
Chocolate	12	43	\$516
Шоколад	23	50	\$1150
σοκολάτα	32	19	\$608
ช็อคโกแลต	14	13	\$182
চক্কোলা	60	12	\$720
चॉकलेट	10	20	\$200
চকোলেট	11	23	\$253
ચોકલેટ	35	35	\$1225
巧克力	21	10	\$210

# pdfHTML: Code sample

```
HtmlConverter.convertToPdf(new File(HTML_FILE_PATH), new File(OUT_FILE_PATH));
```

- ≡ Single line of code
- ≡ Fonts are loaded automatically
- ≡ Integrated support of complex scripts

ស្ករ	60	12	\$720
ចំណី	10	20	\$200
ចំណី	11	23	\$253
ចំណី	35	35	\$1225
ចំណី	21	10	\$210

User Expert

Save

Items

☐ Zero results allowed  
☐ Multiple results allowed

Add selector...

Table

Select row  
Single row like '2' or range like '2:3'  
Select column  
Single column like '2', range like '2:3'  
Column for table building  
Number of column for defining rows

☐ Automatic headers

Format of headers

simple

Headers

Item  
Price  
Quantity  
Amount

Result:

Table: 14 rows, 4 columns

Thank you for your purchase! Below are your contact details

First name

Alexey

Last name

Subach

Email

alexey.subach@itextpdf.com

Purchased items

Item	Price	Quantity	Amount
Chocolate	12	43	\$516
Шоколад	23	50	\$1150
σοκολάτα	32	19	\$608
ช็อคโกแลต	14	13	\$182

Selector	Property	Value			
Table	Headers: Item Price Quantity Amount	Item	Price	Quantity	Amount
		Chocolate	12	43	\$516
		Шоколад	23	50	\$1150
		σοκολάτα	32	19	\$608
		ช็อคโกแลต	14	13	\$182
		சுசுஞா	60	12	\$720
		चॉकलेट	10	20	\$200
		চকোলেট	11	23	\$253
		ચોકલિટ્	35	35	\$1225
		ಚಾಕೊಲೇಟ್	21	10	\$210
		ചോക്കേക്സ്	12	31	\$372
		ଚକୋଲେଟ୍	7	29	\$203
		சாக்லேட்	7	29	\$203

ITEXTDITO

Project Edit Help

Templates

project

invoice-chocolate

Resources

img

itext.png

Template Stylesheets

bootstrap.min.css

embedded

embedded

Template Elements

field

input

rich-text

field

input

rich-text

ITEXT

Invoice form

Thank you for your purchase! Below are your contact details

First name

Last name

Email

Purchased items

Item	Price	Quantity	Amount

Total

Toolbox

Rich text

Image

Subform

Barcode

Header

Text field

Checkbox

Table

Dropdown

Footer

Properties

Base

Label

Last name

Label position

before

Input Type

text

Placeholder

Anchor Id

Static id

Disabled

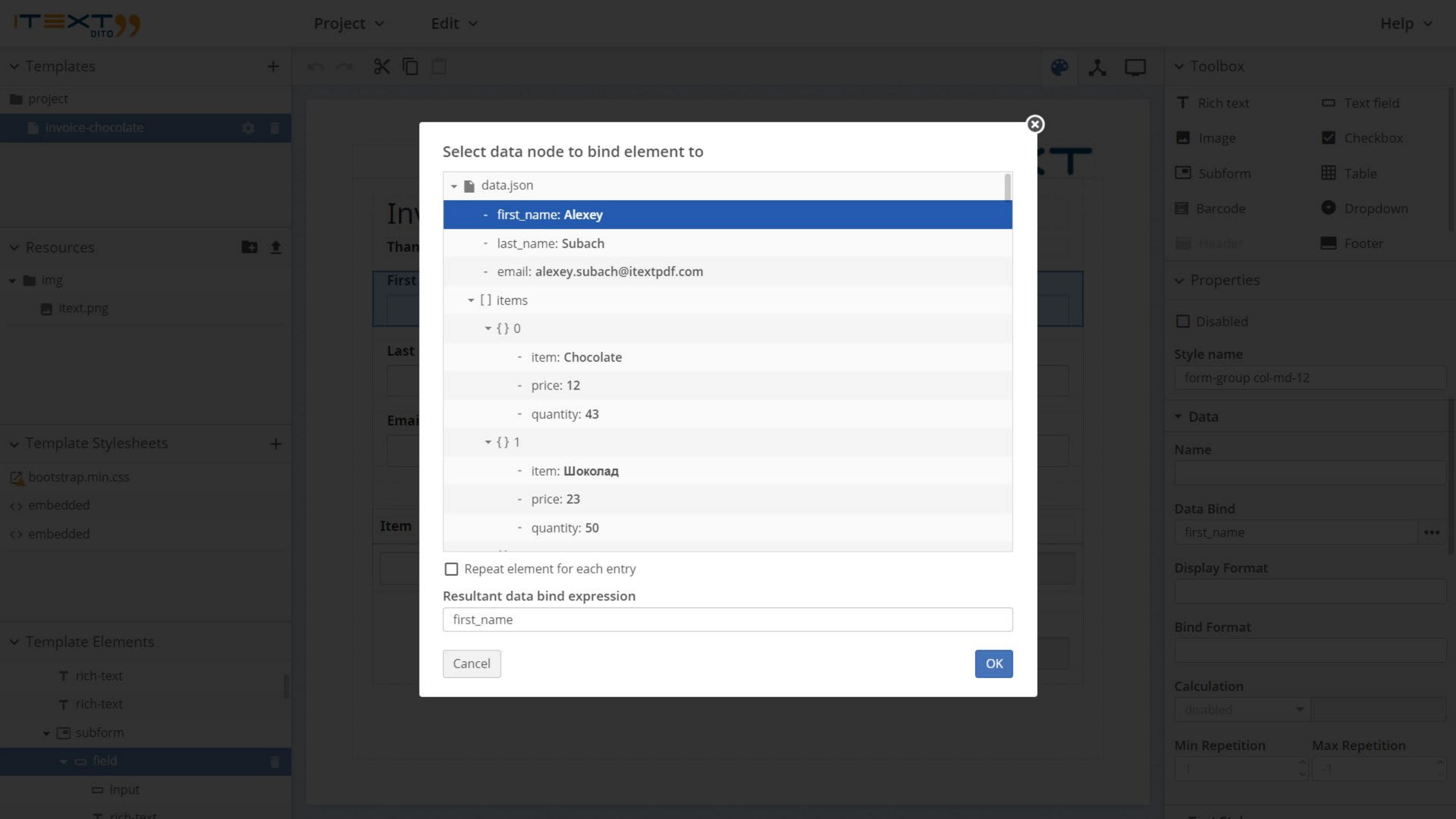
Style name

form-group col-md-12

Data

Name





### Select data node to bind element to

- data.json
  - first\_name: Alexey
  - last\_name: Subach
  - email: alexey.subach@itextpdf.com
  - [] items
    - {} 0
      - item: Chocolate
      - price: 12
      - quantity: 43
    - {} 1
      - item: Шоколад
      - price: 23
      - quantity: 50

☐ Repeat element for each entry

Resultant data bind expression

first\_name

Cancel

OK



# iText DITO: Code sample

```
JsonData data = new JsonData(Files.readString(Path.of(DATA_PATH),  
    StandardCharsets.UTF_8));  
  
try (FileOutputStream fos = new FileOutputStream(OUT_FILE_PATH)) {  
    PdfProducer.convertTemplateFromPackage(new File(TEMPLATE_PROJECT_PATH),  
        "invoice-chocolate", fos, data);  
}
```

**POST** {server-address}/api/pdf-producer

**Content-Type** application/json

**Body**

```
{
  "templateProjectPath": "invoice.dito",
  "templateName": "invoice-chocolate",
  "data": {
    "first_name": "Alexey",
    "last_name": "Subach",
    "email": "alexey.subach@itextpdf.com",
    "items": [
      ...
    ]
  }
  "path": "custom/path/to/file.pdf",
  "properties": { "pdfVersion": "2.0" }
}
```



### PDF Splitter

Split a PDF into two PDFs >



### PDF Merger

Merge two PDFs into one PDF >



### PDF Rotator

Rotate a PDF or specific pages >



### PDF Page Deleter

Delete pages from a PDF >



### PDF Protector

Add password and PDF permission options to a PDF >



### PDF Unlocker

Remove password protection from a PDF >



### Image Converter

Convert an image to PDF >



### HTML Converter

Convert HTML to PDF with pdfHTML >



### XFA Flattener

Flatten an XFA file to static PDF with pdfXFA >

<https://itextpdf.com/demos>

TELEPHONE

Page 1

Value	Screenshot	Correct?
+1 617 982 2646	<div>+1 617 982 2646</div>	<div>OK</div>

TABLE

Page 1

Value					Screenshot	Correct?
	Description	Quantity	Unit price	Discount	Total	<div><div>OK</div></div>
			USD	%	USD	
	Commercial Invisibility Cloak License .NET NO-CAN-SEE-1 Perpetual License - Unlimited use in Time - Unlimited Users	1.00	\$6,800.00	0.00 %	\$6,800.00	
	Updates and Support Commercial Invisibility Cloak .NET NO-CAN-FAIL-1 Technical Support and Updates during 12 Months. Renewed on an Annual basis.	1.00 Year(s)	\$666.00	0.00 %	\$666.00	
	Commercial Non-Production House Elf DBBY-OR-WNKY Perpetual License - Ready for Development - Testing - QA - Training - Disaster Recovery (Passive)	1.00 Server(s)	\$3,140.00	0.00 %	\$3,140.00	
	Updates and Support Commercial Non- Production House Elf THE-CANE Technical Support and Updates during 12 Months	1.00 Year(s)	\$152.00	0.00 %	\$152.00	

Description	Quantity	Unit Price USD	Discount %	Total USD
Commercial Invisibility Cloak License .NET NO-CAN-SEE-1 Perpetual License - Unlimited use in Time - Unlimited Users	1.00	\$6,800.00	0.00 %	\$6,800.00
Updates and Support Commercial Invisibility Cloak .NET NO-CAN-FAIL-1 Technical Support and Updates during 12 Months. Renewed on an Annual basis.	1.00 Year(s)	\$666.00	0.00 %	\$666.00
Commercial Non-Production House Elf DBBY-OR-WNKY Perpetual License - Ready for Development - Testing - QA - Training - Disaster Recovery (Passive)	1.00 Server(s)	\$3,140.00	0.00 %	\$3,140.00
Updates and Support Commercial Non-Production House Elf THE-CANE Technical Support and Updates during 12 Months	1.00 Year(s)	\$152.00	0.00 %	\$152.00

<https://pdf2data.online>

ITEX T DITO

Project ▾ Edit ▾

Schedule a Demo Send Feedback Help ▾

Templates

- transaction\_history
  - Input Templates +
  - Output Templates +
  - output ⚙️ 🗑️

Template Elements

- subform
  - page-header
    - subform
      - table
        - table-colgroup
          - table-col
          - table-col
        - table-body
          - table-row
            - table-cell
              - Image
            - table-cell

Element Data Properties

Data

Name

Data Connections

Samples	+	Conditions
- Fee:	0	
- Net:	-29	
- Balance:	14,113.08	
- Transaction_ID:	5YS82552S0T417...	
- From_Email_Ad...	: services@hotj...	
- Name:	Hotjar Ltd.	
- Bank_Name		
- Bank_Account		
- Shipping_and_H...	0	
- Sales_Tax:	0	
- Invoice_ID		
- Reference_Txn_ID:	B-5BP988A593...	
▼ { } 1		
- Date:	1/7/2019	
- Time:	16:30:16	
- Time Zone:	Europe/Berlin	

Transaction History - EUR

Date	Description	Name \ Email	Currency	Gross	Fee	Net
{{Date}}	{{Description}}	{{Name}} {{From Email Address}}	{{Currency}}	{{Gross}}	{{Fee}}	{{Net}}

<https://itextdito.online>



Thank you! Questions?

[alexey.subach@itextpdf.com](mailto:alexey.subach@itextpdf.com)

<https://itextpdf.com>