# Presentation Creation Guideline

This document describes the **process** of creating course **presentation** (slides) for the educational program at Software University (SoftUni).

## Objectives

* **Learn by examples**
  + Less theory, more “**live examples**” (live code snippets).
  + Avoid explanatory slides, show **more code** instead.
* **Learn by doing**
  + Let the students **do live exercises** in class / repeat the lesson examples themselves.
  + Provide a way for **automated evaluation** for the student’s code (e.g. in a judge system).
* Slides as **textbook** for the **students**
  + Comprehensive slides serve as textbook for the students.
* Slides as **textbook** for the **trainers**
  + Comprehensive slides truncate the preparation before the lessons.

## Presentation Structure

* **Intro** Section
  + **Cover** Slide
  + **Table of Contents** Slide
  + **Sli.do** Slide
* **Content** Sections (3-4 Sections)
  + **Section Cover** Slide
  + **Theory** + Code **Examples** (2-3 Slides)
  + **Problem** Slide + **Solution** Slide (2-3 problems)
* **Practice** (Lab) Slide
  + Optional between the sections
  + At least one such slide per presentation
* **Conclusion** Section
  + **Summary** Slide
  + **Questions** Slide
  + **License** Slide
  + **SoftUni Banners** Slide

## Design Guidelines

* Use **English** (unless you have a good reason to make an exception)
  + English is the primary language for the training resources at SoftUni.
* **Less text** – describe everything in short form
  + Use short bullets, not full sentences.
  + Use very brief descriptions for the problems.
* **Less theory**, more code examples
  + Students **fall** **asleep with theory** (e.g. explain all parameters of certain function).
  + Explain the theory by **code examples** (when possible).
  + Use images / **diagrams** for explaining important concepts.
* Use **code examples**, instead of explanations
  + **Animate** the displaying of the source code if is it long (show the next code section on click).
  + **Visualize** the code results (when appropriate), e.g. show the code execution results.
* **Code examples** everywhere
  + If you have a **slide** about something, you should have **code examples** about it.
  + Theory without code is meaningless à remove it.
* Use **problems** for live coding
  + Provide **problems + solutions** after each significant piece of information à learn by doing.
  + Describe problems very **briefly**.
  + Avoid long descriptions à make them short.
  + Always show **sample input + output**: 1-3 examples.
* Provide **solutions** after each problem
  + Solutions might be fully functional code, or an **essential** **piece of code** (with “TODO” sections).
  + **Animate** the displaying of the source code if is it long (show the next code section on click).
  + **Long code** might be given in external file / resource.
* The **essential** information should be **on the slides**
  + The major lesson content should be on the slides.
  + Long source code examples (demos) might stay in external ZIP archive.
  + You may use **slide notes** for longer examples and long text description.
* **Less information** on a single slide
  + Use **“appear” or “fade” animations** when the slide holds multiple sections.
  + **Split long slides** into several slides.
* **Timing**
  + Use **short presentations** à 20-40 slides
    - Slides: 60-90 minutes
    - Live coding: 30-60 minutes
    - Slides: 30-60 minutes
    - Live coding: 60-90 minutes
  + More than 40 slides à split into multiple presentations

## Presentation Slides

Following the above principles and guidelines, let’s go through a **sample** high-quality course presentation slides.

### Intro Section

|  |  |
| --- | --- |
|  | * The intro section holds the **first few slides** like cover and table of contents. |

#### Cover Slide

|  |  |
| --- | --- |
|  | * Use a short **title** and more descriptive **subtitle**. * Put appropriate **picture**, related to the presentation topic. * Change the **text** above the SoftUni wizard. * Change the **author** when appropriate. |

#### Table of Contents Slide

|  |  |
| --- | --- |
|  | * You may use **topics** and **subtopics**. * Order the topics (1, 2, 3, …). Use bullets for subtopics. * **Animate** the appearance of each topic (with its subtopics). * Use **Appear** or **Fade** animation * Use appropriate **pictures** / **icons**. Find free images / icons in [Google Image Search](https://images.google.com/). * pixabay.com |

#### Sli.do Slide

|  |  |
| --- | --- |
|  | * Create a **Sli.do code** for asking questions. |

### Content Sections (3-4 Sections)

|  |  |
| --- | --- |
|  | * Split the learning content into **content topics**. * Each **content section** holds **a few slides**: section cover + piece of information + code examples + problems + solutions. * The presentation usually holds **4-5** content sections. |

#### Section Cover Slide

|  |  |
| --- | --- |
|  | * Use a short **section topic**. * Use longer section **subtopic**. * Put appropriate **image**. Find free **images** / **icons** in [Google Image Search](https://images.google.com/). Additionally, put some **screenshots** / **code snippets**. |

#### Theory + Code Examples (2-3 Slides)

|  |  |
| --- | --- |
|  | * Prefer **code** instead of **theory**. * Content should be explained by **examples**: code + result of this code. * Avoid **descriptive explanations**. * **Animate** the appearance of the code sections. * Format the **source code** like shown on the slide. * Optionally, add images / icons. |
|  | * Use **balloons** to accent on or explain something specific in the code. * **Colorize** (in yellow) pieces of the code for better **code readability**. * You may use **Calibri** font for the **text** and **comments** in the source code. This saves space. * Never intersect balloons. |
|  | * If the **file names** are important, show them explicitly. * Use **animations** and **arrows** to explain the concepts. * Show also the **output** of your code, e.g. **screenshot**. |

#### Problem Description Slide (2-3 Problems)

|  |  |
| --- | --- |
|  | * Use a slide title “**Problem: Some Problem Title**”. * Describe the problem as **shortly** as possible. * Never use full sentences, just **short bullets**. * Obligatory put a **sample input** + **output**. * Put **several sample inputs** / **outputs** if possible. |

#### Solution of Problem Slide (after each Problem Slide)

|  |  |
| --- | --- |
|  | * Either put **complete source code** (for simple problems) or **portions of the solution** (for bigger code with TODOs inside). * **Animate** the appearance of the code sections. * Provide a sample code invocation if appropriate. * Provide a **link to the judge system**. Link to the entire contest, not a particular problem. This saves time when problem is inserted / deleted. |
|  | * Use “**TODOs**” when the code does not fit in the slide. * Never use font smaller than 20pt. * Prefer font size of 24pt - 28pt. |

### Practice (Lab) Slide

|  |  |
| --- | --- |
|  | * The practice slide says “Students, please **start coding** for 20-40 minutes”. * Use practice slide **1-2 times** in each presentation. * Put practice slide between the **content sections**. * Put **at least one** practice slide at the end of the presentation. * Use appropriate pictures as decorations, e.g. **screenshots** or code snippets. |

### Conclusion Section

|  |  |
| --- | --- |
|  | * The conclusion section holds the **last few slides** like summary and questions. |

#### Summary Slide

|  |  |
| --- | --- |
|  | * **Summarize** the most important from the presentation in one slide. * **Animate** the appearance of each topic (with its content). * Prefer short **code snippets** instead of text summary. |

#### Questions Slide

|  |  |
| --- | --- |
|  | * Make sure the trainer opens **Sli.do** to **answer** the questions given online. |

#### Partners Slide

|  |  |
| --- | --- |
|  | * The partners slides hold the **sponsors’ banners**. * Make sure the banners are **up to date**. |

#### License Slide

|  |  |
| --- | --- |
|  | * In most cases the **license slide** looks just like the shown on the left. * Unless you have a good reason, **leave it unchanged**. * **Acknowledge** any previous work, in case you base your slides content on external resources. |

#### SoftUni Banners Slide

|  |  |
| --- | --- |
|  | * In most cases the **banners slide** looks just like the shown on the left. * Unless you have a good reason, **leave it unchanged**. |

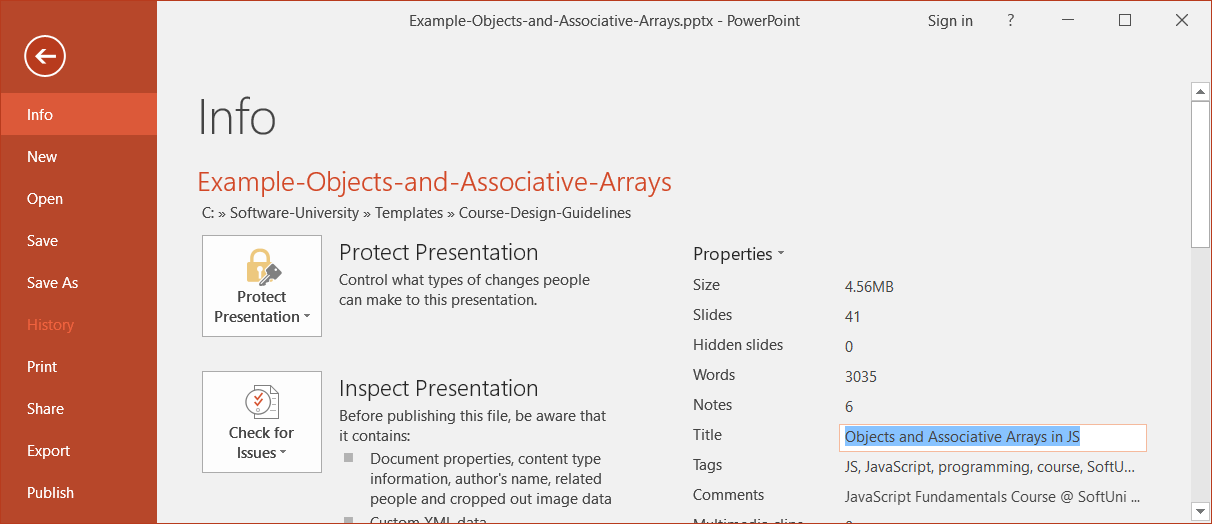
## Formatting Guidelines

These are the SoftUni slides general formatting guidelines.

### Metadata

Fill presentation **metadata**

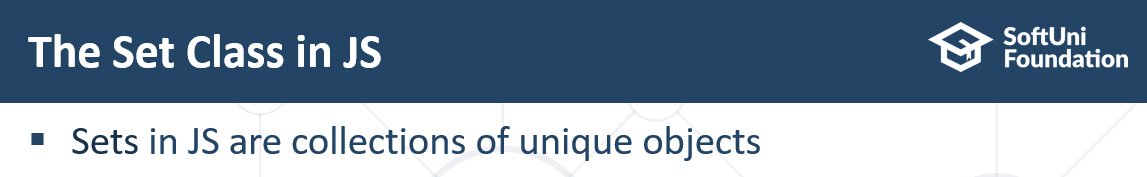
* Fill the **document properties**: title, tags, categories, comments, subject, company, author.
* Maintain **consistent metadata** for the entire course (similar metadata for all course presentations).



* + **Title**: ***{topic title}***
  + **Tags**: ***{course tag 1}***, ***{course tag 2}***, ***{course tag 3}***, …, Software University, SoftUni, programming, coding, software development, education, training, course
  + **Categories**: ***{course category 1}***, ***{course category 2}***, ***{course category 3}***, …, computer programming, programming
    - **Subject**: ***{course name}***  – Practical Training Course @ SoftUni
  + **Comments**: ***{course name}*** Course @ SoftUni – <https://softuni.bg/courses/>... (course URL)
  + **Author**:Software University Foundation

### Slide Titles

* Use short slide **titles** (titles are not sentences).
* **Capitalize Each Slide Title Word** (except prepositions and conjunctions).



### Fonts

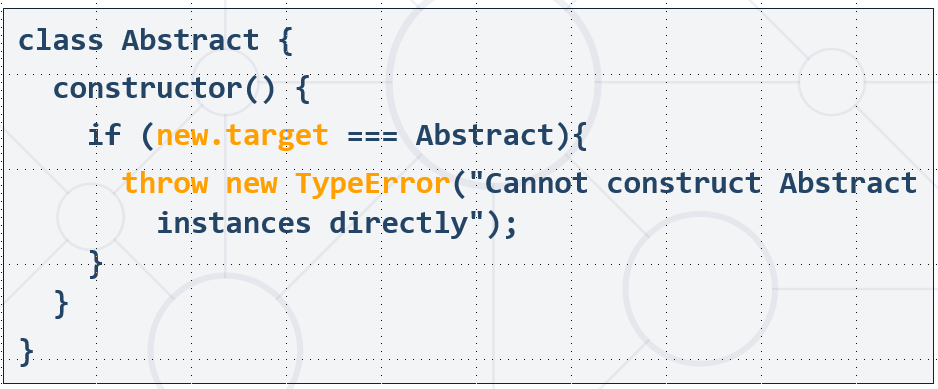
* Use the font sizes coming from the **template** (34pt, 32pt, 30pt).
* Use font sizes in the range [**20pt** … **38pt**].
* Avoid font size smaller than 20pt.
* Use **Calibri** for the **text** on the slides.
* Use font Consolas + **bold** for the **source code** and code elements (like file names).
* Accent by coloring in **yellow** the **key concepts**.

### Colors

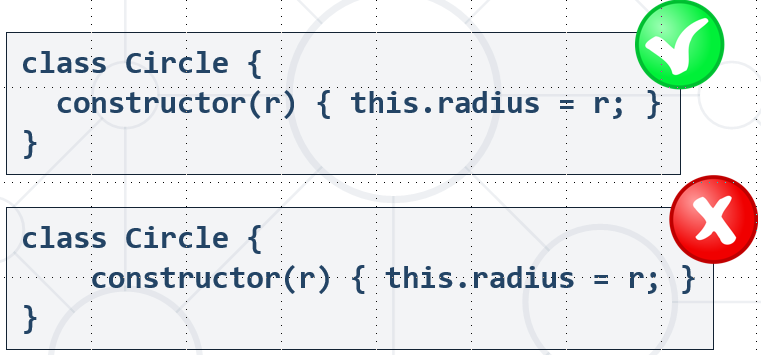
* Use the colors coming from the **template**.
* Never use **low contrast** colors (e.g. red on black background).

### Formatting Code Blocks

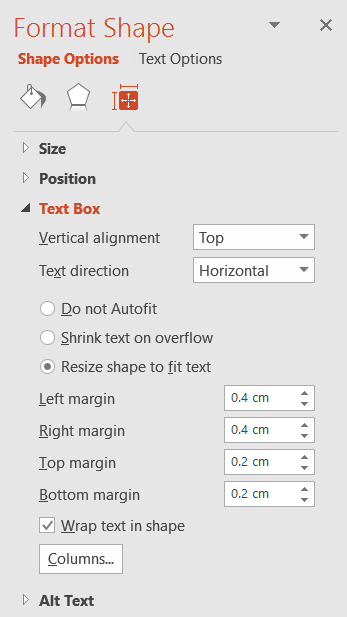
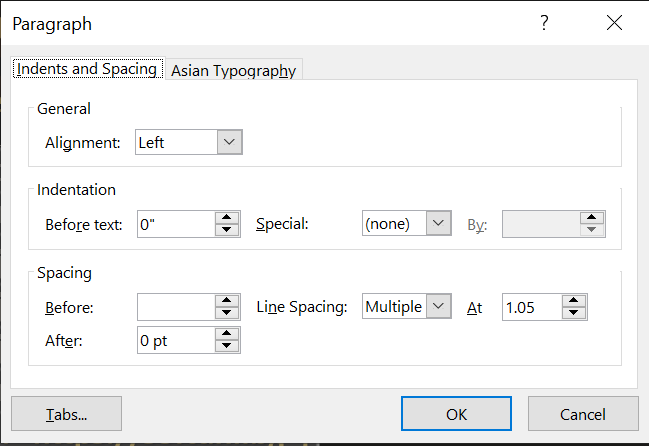
Use the code blocks from the **template**. Don’t try to make a code block yourself!



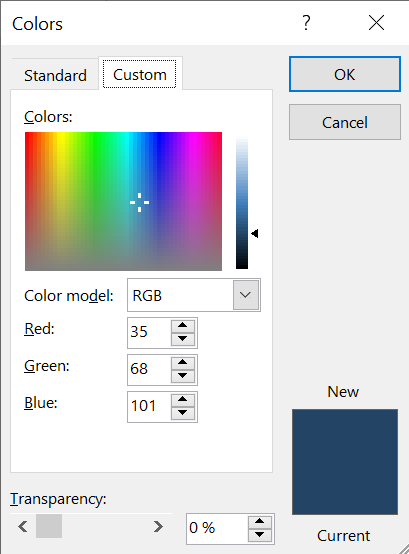
Use **indentation** of **2 spaces**, no more!



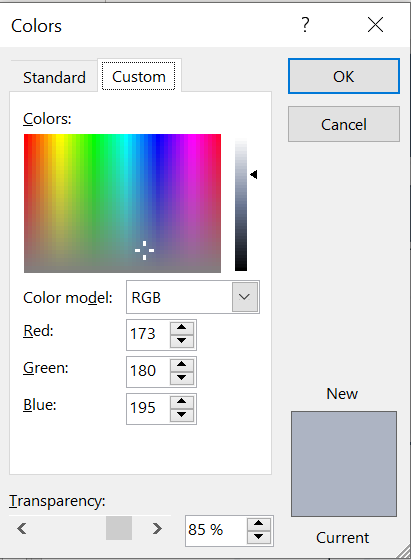
Unless you have a good reason, use the following **margins** for the code blocks:

Use the following **color** for the code, **not white**:



Use the following **color** for the code background, **not white**:



## Typical Mistakes

Let’s review some **typical mistakes** made by the slide authors.

### Too Much Text on the Slide

|  |  |
| --- | --- |
|  | * **Never do this**! So much text makes the student start reading it and stop listening the speaker. * Don’t use complete sentences, use **short bullets**. * If the **information is much**, after sentences were made short, **reduce it**. * Or just **split the slide** into two slides with less content. |
|  | * This is how the above slide can be **fixed**. * **Remove the explanations** à nobody reads them. * **Examples** will describe the problem better. * Give **good examples**. * Describe the problem **very briefly**. The trainer could explain it better in class. |

### Too Much Code on the Slide

|  |  |
| --- | --- |
|  | * Never use font size **less than 24pt**! * Prefer font size **28pt** or **30pt**. * Best solution: reduce the code by **removing unimportant parts** (insert TODOs). * Another solution: **split the code** into several slides. |
|  | * This is the **fixed slide**. * It holds **less code**, with larger, readable **fonts**. * Unimportant pieces of code are replaced with **TODO** comments. |

### Missing Code Example

|  |  |
| --- | --- |
|  | * This slide is **really bad**! * Instead of giving a few **examples** of AND, OR, IN, etc., it just mentions that these predicates exist. * Technical slides should hold **examples**, not mentions. |
|  | * This is how such slides can be fixed: just apply the concept “**learn by examples**”. * Replace the information from the above slides saying “This and this exists” with **code examples** “How to use this and this in practice”. * Focus on **code examples**. People learn best with examples, followed by live coding (problems + solutions + submit in the judge). |

### Font Too Small

|  |  |
| --- | --- |
|  | * This slide is **really bad**! * It uses **too small font**, because the text is too much. |
|  | * This slide is **also bad**! * It holds less text, but **the** **font is still too small**. * The font should be **enlarged** significantly. |
|  | * **Split the slide** into 2-3 smaller slides. * Put **less text** at each slide. * Optionally, add appropriate pictures. |

### One Sentence Slides

|  |  |
| --- | --- |
|  | * This slide is **really bad**! * Instead of giving a **code example** of reducing complexity, it just says something. * If a **new trainer** needs to teach this presentation, he should prepare his own examples and “**invent**” the missing information + sample code. * Even with **notes below the slides**, this code cannot be a good teaching instrument! |
|  | * The above slide completely **redesigned**. * This is how the concept from the previous slide could be **explained by code example**. * Just show a few **lines of code** (as few as possible). * The trainer will have a **real example** and will explain the concept behind the code himself. |

### Inappropriate Colors

|  |  |
| --- | --- |
|  | * The **colors** on the slide are **bad**. * **Low contrast**! * On the **multimedia projector** the green and red colored text will disappear! |
|  | * Improved colors with **higher contrast**. |

### Bad Slide Titles

|  |  |
| --- | --- |
|  | * Slide title is **too long**. * Title **casing** is incorrect à should be “**Capital Case**”. |
|  | * This is the **fixed slide title**. * Now is it **short**. * It **explains** the slide in short. * It is in “**Capital Case**”, i.e. each word starts with capital letter except (a, the, and, or, not, …). |

### Overlapping Content

|  |  |
| --- | --- |
|  | * This is **very bad slide**! * It **cannot be read** without playing the animations inside. * Thus, the slides cannot be used as **textbook** / reference / guide. * It can be **split into several simpler slides** with less animations. |
|  | * Now the slide is **readable**. * **“Appear” or “Fade” animations** still can be used to show the evaluation components one by one. * No overlapping content. |

### Bad Slide Formatting

|  |  |
| --- | --- |
|  | * This slide is **formatted ugly**. * **Code boxes** are too wide. * **Code boxes** have no **margin** from the text. * The **code font** is too small * The **code formatting** is not good. * Тhe **code** can be made **shorter** and cleaner. * The **text is too much** à should be reduced. |
|  | * The same slide **improved**. * **Less text** à **more margin** around it. * Code is **better formatted**, with **coloring**. * Less code, **bigger fonts**. * Appearance **animation** is added for the “iterate over the result collection” + the code box below it. |

### Slides Explaining Methods and Parameters

|  |  |
| --- | --- |
|  | * Slides like this are really **boring**! * Instead of listing the operations and explaining what they do, just **show examples**! |
|  | * This slide is **better**: it explains the operations by **code examples**. * Still, the **content is too much** à the slide can be **split** into two slides. |
|  | * This is how the previous slide can be **split into two slides** with less content. * Such slides have less content and **bigger fonts**. * The **code** **examples** are improved. * The **output** is shown, which improves the understandability (no need to think or guess what will be the result). * The code is **colored** for improved readability. |

### Missing Cover Image

|  |  |
| --- | --- |
|  | * This slide has **missing cover image**. * How to find a good image? * Search for “**database** + **icon**” in Google images: https:// google.bg/search?q=database+icon&tbm=isch * You may combine several icons * Also, the **title** should be “How Do RDBMS Work?” |
|  | * This is how this slide can be **corrected**. * A few **icons** are added to make it look better. * **Titles** are corrected and better formatted. |

### Low Contrast Images

|  |  |
| --- | --- |
|  | * Low contrast image. Use **higher contrast**. * Red on black background will be lost on some multimedia projectors. Just **correct the colors**: |
|  | * This is the **corrected** slide. * Now the colors have **better contrast**. |

### Bad Formatting + Inconsistent Colors

|  |  |
| --- | --- |
|  | * Many problems: * Inconsistent **colors**. * **Balloons** are not the correct form and size. * Small **fonts**. * What is **row** / **column** / **data** is not obvious. * Slide **title** could be better. |
|  | * Now the slide is improved. * **Colors** are the correct one. * **Fonts** are bigger, white text has black **shadow** for better contract. * **Balloons** are the correct ones. * Slide **title** is improved. * Rows / columns / data are **visualized**. * **Animation** is added for rows, columns, etc. |

### Bad Cover Slides

|  |  |
| --- | --- |
|  | * Wrong **title** (typo). * Wrong **formatting**. * Wrong **picture** à not helping to understand the topic. |
|  | * Now the **title** is correct. * A **subtitle** adds more info about the next section. * The **picture** is adequate: explains the concept. * The **formatting** is consistent with the SoftUni slides guidelines. |

### Wrong Margins / Padding

|  |  |
| --- | --- |
|  | * Wrong **margins** around the code blocks * Wrong **positioning**: code blocks should be **centered**. * Wrong **colors**: code should use this color, **not white**! |
|  | * Better **margins**. * Code blocks are **centered**. * **Less text** on the slide, just the important things. * **Bigger text** in the code blocks. * **Colors** in the code blocks are correct (not white). * Important code is **highlighted** (with rectangles). * Added **animation** to show the second part. |

### Over-Descriptive Slide / Too Much Text

|  |  |
| --- | --- |
|  | * **Too much text** on the slide. * The explanations are **over-descriptive**. * Slides are not books, mention **the important information** only. * Avoid full sentences, just **keywords** / **phrases**. |
|  | * Add **pictures** for better visualization and less text. * The same class hierarchy is **visualized**, not described with text. * The **text** is significantly **reduced**.   + Everyone knows how to calculate circle area.   + The next slide shows the toString() output, it does no need to be described. |

### Bad Code Formatting

|  |  |
| --- | --- |
|  | * **Too much code** on the slide. * **Indentation** should be 2, not 4 spaces. * **Font-size** can be increased. * Unneeded {} à can be reduced. * Code block not **centered**. * **Missing spaces** before { and after (). * Incorrect code **color**. |
|  | * **Less code**, bigger **fonts**, bigger **line spacing**. * Used the correct code block **colors**. * Added explanatory **balloon**. * Added **animations** to display the different code pieces one after another (on click). |