# You

\newcommand{\sectionTitleWithImage}[2]{
\begin{minipage}[t]{0.05\textwidth} % Adjust width as needed
\includegraphics[width=1cm height=1cm]{#1}
\end{minipage}
\hfill
\begin{minipage}[t]{0.9\textwidth} % Adjust width as needed
{\section{\texorpdfstring{\color{accentcolor}\enspace \textsc{#2}}{#2}}}
\end{minipage}
}
In the above I find text like height on left of the image. Can you simply above without using minipage, how I can include image on the left and text on the rigth with image aligning with text?

# Copilot

Sure, you can use the \raisebox command to adjust the vertical position of the image relative to the text. Here's a simplified version of your command:

\newcommand{\sectionTitleWithImage}[2]{

\includegraphics[width=1cm, height=1cm]{#1}

\hspace{0.5cm} % Adjust space as needed

\section{\texorpdfstring{\color{accentcolor}\enspace \textsc{#2}}{#2}}

}

In this version, the image and the section title are on the same line. The \hspace command is used to add some space between the image and the text. You can adjust the space as needed. If you find that the image is not aligning well with the text, you can use the \raisebox command to adjust the vertical position of the image. Here's how you can do it:

\newcommand{\sectionTitleWithImage}[2]{

\raisebox{-0.5\height}{\includegraphics[width=1cm, height=1cm]{#1}}

\hspace{0.5cm} % Adjust space as needed

\section{\texorpdfstring{\color{accentcolor}\enspace \textsc{#2}}{#2}}

}

In the \raisebox command, -0.5\height adjusts the vertical position of the image. You can change this value to suit your needs. The negative value will lower the image, and the positive value will raise it. The \height refers to the height of the image. So -0.5\height will lower the image by half of its height. You can adjust this value to get the alignment you want. Please note that you need to use the graphicx package to use the \includegraphics and \raisebox commands. You can add \usepackage{graphicx} in the preamble of your document if it's not already there. I hope this helps! Let me know if you have any other questions. 😊

# You

Image is above text when I use:

\newcommand{\sectionTitleWithImage}[2]{
\raisebox{-0.5\height}{\includegraphics[width=0.75cm, height=0.75cm]{#1}}
\hspace{0.5cm} % Adjust space as needed
\section{\texorpdfstring{\color{accentcolor}\enspace \textsc{#2}}{#2}}
}

It should be inline with text but on left of text with some space in between like \enspace

# Copilot

I apologize for the confusion. The \raisebox command adjusts the vertical position of the image. The -0.5\height argument is lowering the image by half of its height. If the image is appearing above the text, you can try adjusting this value.

However, if you want the image to be inline with the text, you might not need the \raisebox command at all. Here's a simplified version of your command without \raisebox:

\newcommand{\sectionTitleWithImage}[2]{

\includegraphics[width=0.75cm, height=0.75cm]{#1}

\hspace{0.5cm} % Adjust space as needed

\section{\texorpdfstring{\color{accentcolor}\enspace \textsc{#2}}{#2}}

}

In this version, the image and the section title are on the same line. The \hspace command is used to add some space between the image and the text. You can adjust the space as needed.

If the image is still not aligning well with the text, it could be due to the specific properties of the image file or the text. In that case, you might need to adjust the image file or use additional LaTeX commands to get the desired alignment. I hope this helps! Let me know if you have any other questions. 😊

# You

Image at #1 is way above text #2.

# Copilot

I apologize for the confusion earlier. The \raisebox command in LaTeX allows you to raise or lower text. The -0.5\height argument is lowering the image by half of its height. If the image is appearing above the text, you can try adjusting this value.

Here's a revised version of your command:

\newcommand{\sectionTitleWithImage}[2]{

\raisebox{-0.25\height}{\includegraphics[width=0.75cm, height=0.75cm]{#1}}

\hspace{0.5cm} % Adjust space as needed

\section{\texorpdfstring{\color{accentcolor}\enspace \textsc{#2}}{#2}}

}

In the \raisebox command, -0.25\height adjusts the vertical position of the image. You can change this value to suit your needs. The negative value will lower the image, and the positive value will raise it. The \height refers to the height of the image. So -0.25\height will lower the image by a quarter of its height. You can adjust this value to get the alignment you want.

Please note that the exact alignment may depend on the specific properties of the image file and the text. If the image is still not aligning well with the text, you might need to adjust the image file or use additional LaTeX commands to get the desired alignment. I hope this helps! Let me know if you have any other questions. 😊

# You

I have a document named documentMETADATA.cls and other tex documents. I want to override font sent cls file and apply like this:

# Copilot

Sure, you can override the font settings in your .cls file by using the \renewcommand command in your .tex document. Here's an example of how you can do it:

\documentclass{documentMETADATA} % Use your custom class

\begin{document}

% Override the font settings here

\renewcommand{\familydefault}{\sfdefault} % Change to sans-serif font

\renewcommand{\seriesdefault}{\bfdefault} % Change to bold font

\renewcommand{\shapedefault}{\itdefault} % Change to italic font

% Your document content goes here

\end{document}

In this example, the \renewcommand command is used to redefine the \familydefault, \seriesdefault, and \shapedefault commands, which control the default font family, series, and shape, respectively. You can replace \sfdefault, \bfdefault, and \itdefault with the names of the font, series, and shape you want to use.

Please note that the exact commands and names you need to use may depend on the specific font, series, and shape you want to use, as well as the specific settings in your .cls file. If you're not sure about the names of the fonts, series, or shapes, you can look them up in the LaTeX documentation or in the documentation of the specific packages you're using.

I hope this helps! Let me know if you have any other questions. 😊

# You

Can I use:
{\fontfamily{phv}\selectfont}

# Copilot

Yes, you can use {\fontfamily{phv}\selectfont} to change the font. This command changes the font to Helvetica, which is represented by phv in LaTeX. Here's how you can use it in your document:

\documentclass{documentMETADATA} % Use your custom class

\begin{document}

% Override the font settings here

{\fontfamily{phv}\selectfont

% Your document content goes here

}

\end{document}

In this example, the {\fontfamily{phv}\selectfont} command changes the font to Helvetica for the text that is within the curly braces {}. If you want the entire document to be in Helvetica, you can place all your text within these curly braces.

Please note that you need to use the fontenc package with the T1 option to use Helvetica and other PostScript fonts. You can add \usepackage[T1]{fontenc} in the preamble of your document if it's not already there.

I hope this helps! Let me know if you have any other questions. 😊

# You

I saw that in tex file, the document class is like this:

\documentclass[localFont,alternative]{documentMETADATA}
What do you suggest?

# Copilot

The \documentclass[localFont,alternative]{documentMETADATA} command indicates that the documentMETADATA.cls class file is being used with the localFont and alternative options. These options are likely defined within the documentMETADATA.cls file and will change the behavior or appearance of the document in some way.

If you want to override the font to Helvetica in your .tex file, you can still do so using the \fontfamily{phv}\selectfont command. However, you should place this command after the \begin{document} command to ensure that it overrides any font settings specified by the localFont option. Here's how you can do it:

\documentclass[localFont,alternative]{documentMETADATA} % Use your custom class

\begin{document}

% Override the font settings here

{\fontfamily{phv}\selectfont

% Your document content goes here

}

\end{document}

In this example, the {\fontfamily{phv}\selectfont} command changes the font to Helvetica for the text that is within the curly braces {}. If you want the entire document to be in Helvetica, you can place all your text within these curly braces.

Please note that you need to use the fontenc package with the T1 option to use Helvetica and other PostScript fonts. You can add \usepackage[T1]{fontenc} in the preamble of your document if it's not already there.

I hope this helps! Let me know if you have any other questions. 😊

# You

Can I not change somewhere in class object because doing like above disturbs all font configurations like bold fonts are converted to non-bold, etc.

Can we not address the issue in DocumentMETADATA.cls file? By tweaking LocalFont, alternative - these options?

# Copilot

Yes, you can definitely make changes directly in the DocumentMETADATA.cls file. This would be a good approach if you want to maintain consistent font settings across multiple documents that use this class.

The localFont and alternative options are likely defined within the DocumentMETADATA.cls file. You would need to locate where these options are defined in the .cls file and modify the font settings there.

For example, if localFont is defined something like this:

\DeclareOption{localFont}{

\renewcommand{\familydefault}{\sfdefault}

% other font settings...

}

You can change the \sfdefault to phv to use Helvetica:

\DeclareOption{localFont}{

\renewcommand{\familydefault}{phv}

% other font settings...

}

Please note that the exact changes you need to make will depend on how localFont and alternative are defined in your .cls file. If you're not sure how to make these changes, you might want to consult the documentation for the class or ask the author of the class for help.

Remember to save the .cls file and recompile your .tex document after making these changes. The new font settings should then be applied to your document.

I hope this helps! Let me know if you have any other questions. 😊

# You

Would your response change if I give below the local font definition, documentMETADATA.cls file:

% Setup fonts
\defaultfontfeatures{Ligatures=TeX}
%

% Define font to use according to localFont option
\ifundef{@local}
{
%\setmainfont[BoldFont = Helvetica Neue, ItalicFont=Helvetica Neue Thin Italic ,SmallCapsFont = Helvetica Neue Light]{Helvetica Neue Thin}
\setmainfont{Source Sans Pro Light}[
BoldFont = SourceSansPro-Regular,
ItalicFont= Source Sans Pro Light Italic]
}
{
\setmainfont{SourceSansPro-Light}[
Path = fonts/,
BoldFont = SourceSansPro-Regular,
ItalicFont = SourceSansPro-LightIt]
}

Also consider following:

\DeclareOption{localFont}{
\def@local{local}
}

\DeclareOption{alternative}{
\def@alternative{alternate}
}

# Copilot

Thank you for providing the additional context. Given this information, you can change the font to Helvetica by modifying the \setmainfont command in the localFont option in your DocumentMETADATA.cls file.

Here's how you can do it:

% Define font to use according to localFont option

\ifundef{\@local}

{

\setmainfont{Helvetica}[

BoldFont = Helvetica-Bold,

ItalicFont= Helvetica-Oblique]

}

{

\setmainfont{Helvetica}[

Path = fonts/,

BoldFont = Helvetica-Bold,

ItalicFont = Helvetica-Oblique]

}

In this example, I've replaced Source Sans Pro Light and SourceSansPro-Light with Helvetica, and SourceSansPro-Regular and SourceSansPro-LightIt with Helvetica-Bold and Helvetica-Oblique respectively. Please note that you need to have the Helvetica font installed on your system and the font names (like Helvetica-Bold and Helvetica-Oblique) might be different based on your system's font installation.

Remember to save the .cls file and recompile your .tex document after making these changes. The new font settings should then be applied to your document.

Please note that the exact changes you need to make will depend on how localFont and alternative are defined in your .cls file. If you're not sure how to make these changes, you might want to consult the documentation for the class or ask the author of the class for help.

I hope this helps! Let me know if you have any other questions. 😊

# You

But my original change was about font family phv.

# Copilot

I apologize for the confusion. If you want to use the phv font family, which corresponds to Helvetica in LaTeX, you can modify the \setmainfont command in your DocumentMETADATA.cls file as follows:

% Define font to use according to localFont option

\ifundef{\@local}

{

\setmainfont{phv}[

BoldFont = phv,

ItalicFont= phv]

}

{

\setmainfont{phv}[

Path = fonts/,

BoldFont = phv,

ItalicFont = phv]

}

In this example, I've replaced Source Sans Pro Light and SourceSansPro-Light with phv, and SourceSansPro-Regular and SourceSansPro-LightIt with phv respectively.

Please note that you need to have the Helvetica font installed on your system and the font names (like phv) might be different based on your system's font installation.

Remember to save the .cls file and recompile your .tex document after making these changes. The new font settings should then be applied to your document.

Please note that the exact changes you need to make will depend on how localFont and alternative are defined in your .cls file. If you're not sure how to make these changes, you might want to consult the documentation for the class or ask the author of the class for help.

I hope this helps! Let me know if you have any other questions. 😊

# You

That caused all errors and non text is seen and \fa symbols? phv is a font family. Have you messed up using that as a font instead of font family definition above?

# Copilot

I apologize for the confusion. You're correct that phv is a font family, and it corresponds to Helvetica in LaTeX. The \setmainfont command is part of the fontspec package, which is used with XeLaTeX and LuaLaTeX, not with pdfLaTeX. The phv font family is used with pdfLaTeX, not with XeLaTeX or LuaLaTeX.

If you're using pdfLaTeX, you can use the helvet package to change the default font to Helvetica. Here's how you can do it:

\usepackage[scaled]{helvet}

\renewcommand\*\familydefault{\sfdefault}

This code changes the default font to a scaled version of Helvetica. The \sfdefault command sets the default font family to sans-serif, and the helvet package changes the default sans-serif font to Helvetica.

If you're using XeLaTeX or LuaLaTeX and want to use Helvetica, you need to use the name of the Helvetica font installed on your system with the \setmainfont command. The name might be something like Helvetica Neue or Helvetica LT Std, but it depends on the specific font installation.

I hope this helps! Let me know if you have any other questions. 😊

# You

What about my code definition which you changed and proved to be wrong?

% Define font to use according to localFont option
\ifundef{@local}
{
\setmainfont{phv}[
BoldFont = phv,
ItalicFont= phv]
}
{
\setmainfont{phv}[
Path = fonts/,
BoldFont = phv,
ItalicFont = phv]
}

# Copilot

I apologize for the confusion earlier. The phv is indeed a font family identifier for Helvetica in the pdftex engine, but it's not the correct way to set fonts when using fontspec with XeLaTeX or LuaLaTeX. The fontspec package expects the actual name of the font, not the pdftex font family identifier.

Given your original code, if you want to change the font to Helvetica, you would need to know the exact name of the Helvetica font installed on your system. For example, if you have the font "Helvetica Neue" installed, you could change your code to:

% Define font to use according to localFont option

\ifundef{\@local}

{

\setmainfont{Helvetica Neue}[

BoldFont = Helvetica Neue,

ItalicFont= Helvetica Neue Italic]

}

{

\setmainfont{Helvetica Neue}[

Path = fonts/,

BoldFont = Helvetica Neue,

ItalicFont = Helvetica Neue Italic]

}

Please replace "Helvetica Neue" with the exact name of the Helvetica font installed on your system. The names for the bold and italic variants might also be different and should be replaced accordingly.

I hope this clears up the confusion. Let me know if you have any other questions! 😊

# You

I can see in overleaf that compiler is lualatex. Changing as above caused all sorts of compilation errors.

# Copilot

I apologize for the confusion earlier. When using LuaLaTeX with the fontspec package, you should use the actual name of the font installed on your system, not the font family identifier used by pdfLaTeX.

If you want to use Helvetica, you need to know the exact name of the Helvetica font installed on your system. The name might be something like "Helvetica Neue" or "Helvetica LT Std", but it depends on the specific font installation.

Here's how you can modify your code:

% Define font to use according to localFont option

\ifundef{\@local}

{

\setmainfont{Helvetica Neue}[

BoldFont = Helvetica Neue Bold,

ItalicFont= Helvetica Neue Italic]

}

{

\setmainfont{Helvetica Neue}[

Path = fonts/,

BoldFont = Helvetica Neue Bold,

ItalicFont = Helvetica Neue Italic]

}

Please replace "Helvetica Neue", "Helvetica Neue Bold", and "Helvetica Neue Italic" with the exact names of the regular, bold, and italic variants of the Helvetica font installed on your system.

If you're still encountering errors, they might be due to other issues in your document or .cls file. If you could provide more information about the errors, I might be able to help you troubleshoot them.

I hope this helps! Let me know if you have any other questions. 😊

# You

What are the alternatives for using regular, bold and italics fonts for the below:

\setmainfont{Source Sans Pro Light}[
BoldFont = SourceSansPro-Regular,
ItalicFont= Source Sans Pro Light Italic]
}
{
\setmainfont{SourceSansPro-Light}[
Path = fonts/,
BoldFont = SourceSansPro-Regular,
ItalicFont = SourceSansPro-LightIt]
}

# Copilot

The alternatives for using regular, bold, and italics fonts depend on the fonts installed on your system. If you want to use a different font, you need to know the exact name of the font and its variants (regular, bold, italic) installed on your system.

For example, if you have the "Arial" font installed on your system, you can change your code to:

\setmainfont{Arial}[

BoldFont = Arial Bold,

ItalicFont= Arial Italic]

And

\setmainfont{Arial}[

Path = fonts/,

BoldFont = Arial Bold,

ItalicFont = Arial Italic]

Please replace "Arial", "Arial Bold", and "Arial Italic" with the exact names of the regular, bold, and italic variants of the font you want to use.

If you're not sure about the names of the fonts installed on your system, you can check the font book (on macOS), the font manager (on Linux), or the Fonts folder in the Control Panel (on Windows).

I hope this helps! Let me know if you have any other questions. 😊

# You

Why the two sections after:

\ifundef{@local}

Like:

(1)

{
%\setmainfont[BoldFont = Helvetica Neue, ItalicFont=Helvetica Neue Thin Italic ,SmallCapsFont = Helvetica Neue Light]{Helvetica Neue Thin}
\setmainfont{SourceSansPro-Black}[
Path = fonts/,
BoldFont = SourceSansPro-Bold,
ItalicFont= SourceSansPro-BlackIt]
}

(2)

{
\setmainfont{SourceSansPro-Black}[
Path = fonts/,
BoldFont = SourceSansPro-Bold,
ItalicFont = SourceSansPro-BlackIt]
}

They look quite similar.

# Copilot

The \ifundef command in LaTeX checks if a command has been defined or not. In your case, it's checking if the command \@local has been defined.

The two sections you're seeing are the two possible branches of this condition:

1. The first section (between the first pair of {}) is the code that gets executed if \@local is **not** defined.
2. The second section (between the second pair of {}) is the code that gets executed if \@local **is** defined.

In your case, both sections are indeed very similar. They both set the main font to SourceSansPro-Black, with SourceSansPro-Bold as the bold font and SourceSansPro-BlackIt as the italic font. The only difference is that in the second section, the fonts are loaded from a specific path (fonts/), which is specified by the Path = fonts/ option. This might be necessary if the fonts are not installed system-wide and are instead included with the document in a local directory.

If the two sections are meant to do the same thing, they could be simplified into a single command outside of the \ifundef condition. However, without more context about the rest of the .cls file and how it's used, it's hard to say why it was written this way. It's possible that there used to be more differences between the two sections, and they were made more similar over time due to changes in the document requirements or the available fonts. 😊

# You

What about this:

\DeclareOption{alternative}{
\def@alternative{alternate}
}

Is it also related to font?

# Copilot

The \DeclareOption{alternative}{...} command in LaTeX is used to define a new option that can be passed to the document class. In this case, an option named alternative is being defined.

When the alternative option is used, the command \def\@alternative{alternate} is executed. This command defines a new command named \@alternative and sets its value to alternate.

This alternative option could potentially be related to font settings if it's used in that context elsewhere in the .cls file. However, based on the code snippet you've provided, it's not directly setting any font options. It's simply defining a new command \@alternative.

To determine how this option affects the font or other formatting of your document, you would need to look for where \@alternative is used in the .cls file or in your .tex document. If it's used in relation to font settings, then yes, it could indirectly be related to the font. Otherwise, it might be used for some other formatting or structuring purposes in your document. 😊

# You

See alternative is referred here:

\newcommand\resumetitle[1]{
\ifundef{@alternative}{
\par{
\bigskip\center{\Large \color{accentcolor}\textbf{#1}\color{Black}}\par
}
\bigskip
}{
\color{accentcolor}\textbf{\large{#1}}\color{black}
}
}

# Copilot

Thank you for sharing this additional context. The \resumetitle command you've shown here uses the \@alternative command that is defined when the alternative option is used in your document class.

The \ifundef{\@alternative}{...}{...} command checks if \@alternative is defined. If \@alternative is not defined (i.e., the alternative option was not used), then it executes the first block of code. This block of code formats the resume title (#1) as centered, large, bold, and in the accent color, followed by a big skip of vertical space.

If \@alternative is defined (i.e., the alternative option was used), then it executes the second block of code. This block of code formats the resume title (#1) as large, bold, and in the accent color.

So, the alternative option in your document class changes the formatting of the resume title in your document. It doesn't directly change the font, but it does change the size and color of the text. If you want to change the font in addition to the size and color, you would need to modify the \resumetitle command or the \@alternative command accordingly. 😊

# You

Instead of using ifundef:

if I have a command like this:

\DeclareOption{localFont}{
\def@local{local}
}

Can I define localFont? How?

# Copilot

The \DeclareOption{localFont}{...} command in your .cls file is defining an option named localFont for your document class. When you use the localFont option in your .tex file, it executes the code inside the {...}, which in this case is \def\@local{local}.

To use the localFont option, you need to include it in the \documentclass command in your .tex file like this:

\documentclass[localFont]{documentMETADATA}

This line of code tells LaTeX to use the documentMETADATA document class with the localFont option. When LaTeX sees the localFont option, it executes the code defined for that option in the .cls file, which is \def\@local{local}.

So, you don't need to define localFont in your .tex file. It's already defined in your .cls file, and you can use it in your .tex file by including it in the \documentclass command.

I hope this helps! Let me know if you have any other questions. 😊

# You

But what is localFont? We need to define that class, isn't it?

# Copilot

The localFont is not a class, but an option that you've declared in your custom class file documentMETADATA.cls. When you use \documentclass[localFont]{documentMETADATA} in your .tex file, it triggers the code associated with the localFont option in your class file.

In your class file, you've associated the localFont option with the command \def\@local{local}. This means that when you use the localFont option, it defines a new command \@local and sets its value to local.

The actual effect of using the localFont option depends on how the \@local command is used elsewhere in your class file. For example, it could be used to conditionally change the font or other formatting settings based on whether the localFont option is used.

So, you don't need to define localFont as a class. It's an option that you've defined in your class file, and you can use it in your .tex file by including it in the \documentclass command. 😊

# You

How can I use @local? Like this:

@local
{
%\setmainfont[BoldFont = Helvetica Neue, ItalicFont=Helvetica Neue Thin Italic ,SmallCapsFont = Helvetica Neue Light]{Helvetica Neue Thin}
\setmainfont{SourceSansPro-Black}[
Path = fonts/,
BoldFont = SourceSansPro-Bold,
ItalicFont= SourceSansPro-BlackIt]
}

# Copilot

The \@local command as defined in your class file is not a command that takes arguments or contains a block of code. Instead, it's a simple command that just holds the string local. So, you can't use it like you've shown in your example.

However, you can use \@local in conditional statements to check if the localFont option has been used. For example:

\ifdefined\@local

% The localFont option was used

\setmainfont{SourceSansPro-Black}[

Path = fonts/,

BoldFont = SourceSansPro-Bold,

ItalicFont= SourceSansPro-BlackIt]

\else

% The localFont option was not used

% Set the font to something else

\fi

In this example, the \ifdefined command checks if \@local is defined. If it is (i.e., if the localFont option was used), then it sets the main font to SourceSansPro-Black. If it's not, then it could set the font to something else.

I hope this helps! Let me know if you have any other questions. 😊

# You

Without using:

\DeclareOption{localFont}{
\def@local{local}
}

How to define localFont in cls file?

# Copilot

The \DeclareOption command is typically used in class files to define options that can be passed to the \documentclass command. If you don't want to use \DeclareOption to define localFont, you could instead define localFont as a new command using \newcommand or \def.

However, please note that this would change how localFont is used. Instead of being an option that's passed to \documentclass, it would be a command that you use in the body of your document.

Here's an example of how you could define localFont as a new command:

\newcommand{\localFont}{

% Define what localFont does here

\setmainfont{SourceSansPro-Black}[

Path = fonts/,

BoldFont = SourceSansPro-Bold,

ItalicFont= SourceSansPro-BlackIt]

}

In this example, \localFont is a command that changes the main font to SourceSansPro-Black. You can use this command in your document like this: \localFont.

Please note that this is just an example. The actual code you need to write will depend on what you want localFont to do. 😊

# You

How about using @def for defining localFont?

# Copilot

I believe there might be a misunderstanding. In LaTeX, there isn't a command called @def. The \def command is a TeX primitive that's used to define a new command, and the @ symbol is used in command names to denote internal commands that are not intended for use in the document.

If you want to define localFont as a new command, you can use the \def or \newcommand command. Here's an example:

\def\localFont{%

% Define what localFont does here

\setmainfont{SourceSansPro-Black}[

Path = fonts/,

BoldFont = SourceSansPro-Bold,

ItalicFont= SourceSansPro-BlackIt]

}

In this example, \localFont is a command that changes the main font to SourceSansPro-Black. You can use this command in your document like this: \localFont.

Please note that this is just an example. The actual code you need to write will depend on what you want localFont to do. 😊

# You

In the below:

\ifundef{@local}
{
% %\setmainfont[BoldFont = Helvetica Neue, %ItalicFont=Helvetica Neue Thin Italic ,SmallCapsFont = %Helvetica Neue Light]{Helvetica Neue Thin}
\def@local{\setmainfont{SourceSansPro-Black}}[
Path = fonts/,
RegularFont = SourceSansPro-Regular
BoldFont = SourceSansPro-Bold,
ItalicFont = SourceSansPro-It]
}
In the sections tex file, everything appears in bold even though I have defined RegularFont.

I have used like this in cv.tex file:
\documentclass[localFont,alternative]{documentMETADATA}

In that file there are references to several sections file:

\input{section\_headline} % Research Statement

\input{section\_experience} % Section Professional Experience

\input{section\_research\_experience} % Section Research Experience

\input{section\_education} % Section Education

\input{section\_certifications} % Section Certifications

\input{section\_tech\_stack} % Section Skills

\input{section\_languages} % Section languaqes

\input{section\_interests} % Section interests

\input {section\_references} % References

# Copilot

I see that you're trying to set the main font to SourceSansPro-Black with SourceSansPro-Regular as the regular font, SourceSansPro-Bold as the bold font, and SourceSansPro-It as the italic font. However, it seems like there's a small mistake in your code which is causing everything to appear in bold.

The \setmainfont command should be followed by a pair of braces {} that contain the name of the font, and then a pair of square brackets [] that contain the font features. In your code, the \def\@local command is inside the braces {} that should contain the name of the font. This is likely causing the issue.

Here's how you can correct it:

\ifundef{\@local}

{

\def\@local{local}

\setmainfont{SourceSansPro-Black}[

Path = fonts/,

RegularFont = SourceSansPro-Regular,

BoldFont = SourceSansPro-Bold,

ItalicFont = SourceSansPro-It]

}

In this corrected version, the \def\@local{local} command is outside the braces {} that contain the name of the font. This should fix the issue and make the RegularFont, BoldFont, and ItalicFont options work as expected.

I hope this helps! Let me know if you have any other questions. 😊

# You

See below:

\ifundef{@local}
{
% %\setmainfont[BoldFont = Helvetica Neue, %ItalicFont=Helvetica Neue Thin Italic ,SmallCapsFont = %Helvetica Neue Light]{Helvetica Neue Thin}
\def@local{local}\setmainfont{SourceSansPro-Black}[
Path = fonts/,
RegularFont = SourceSansPro-Regular
BoldFont = SourceSansPro-Bold,
ItalicFont = SourceSansPro-It]
}

The changes did not produce expected results. Still all fonts are shown as bold. See the image.

# Copilot

I apologize for the confusion earlier. The issue might be due to the font selection in the \setmainfont command. The SourceSansPro-Black font is a heavier weight than SourceSansPro-Regular, which could be why all the text appears bold.

You might want to set the main font to SourceSansPro-Regular instead of SourceSansPro-Black. Here's how you can modify your code:

\ifundef{\@local}

{

\def\@local{local}\setmainfont{SourceSansPro-Regular}[

Path = fonts/,

RegularFont = SourceSansPro-Regular,

BoldFont = SourceSansPro-Bold,

ItalicFont = SourceSansPro-It]

}

In this example, the main font is set to SourceSansPro-Regular, which should make the regular text appear in normal weight. The bold and italic variants are set to SourceSansPro-Bold and SourceSansPro-It, respectively.

Please try this modification and see if it resolves the issue. If not, there might be other factors at play, such as other font settings in your document or .cls file, or the specific properties of the SourceSansPro fonts. If you continue to have issues, please let me know! 😊

Source(s)

1. <https://www.fishbowlapp.com/post/goldman-sachs-early-careers-investing-application-portal-said-interview-since-july-2021-and-has-now-changed-to-under>

2. <https://www.nairaland.com/1064724/studying-finland-free-tuition/346>

3. <https://webronza.asahi.com/science/articles/2020112900003.html>

4. <https://www.levels.fyi/blog/amazon-application-process.html>

5. [https://corp.zozo.com/en/news/20210715-19496](https://corp.zozo.com/en/news/20210715-19496/)

6. <https://www.somaiya.edu/en/view-announcement/354>

7. [https://nvshq.org/time-table/west-bengal-hs-routine](https://nvshq.org/time-table/west-bengal-hs-routine/)