

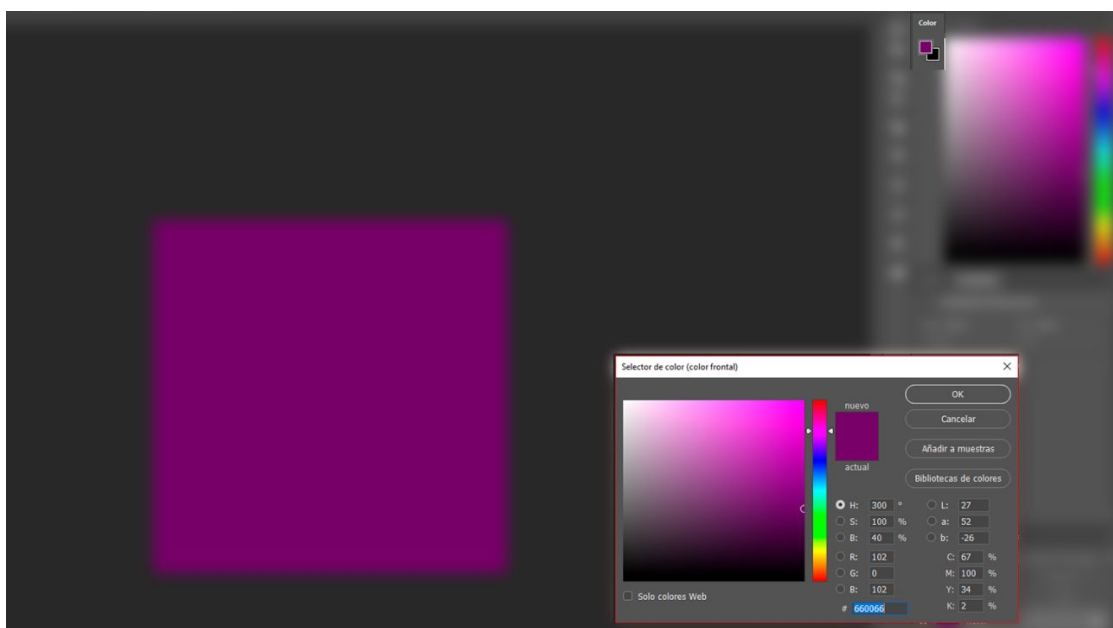
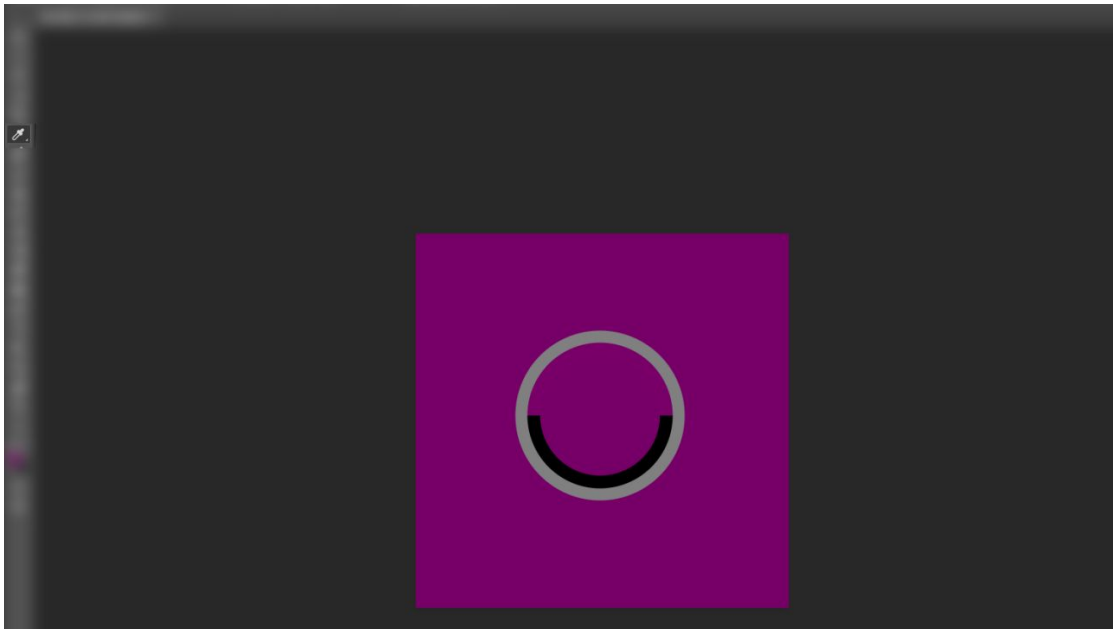
Hex editing colors

So, you've noticed that a lot of the in-game clothes and accessories have parts that aren't colorable, or maybe you want a specific color that the limited customization palette can't give you, or maybe you even want to change the default colors on 1P or 2P outfits. Well, for any of that, we have hex editing!

First step - Choosing a color

Use this site to pick the color you want: https://www.w3schools.com/colors/colors_picker.asp

Use "Pick a color" to select a tone, and then copy the "Lighter / Darker" value on the right you want depending on your desired brightness. You can also use Photoshop to select a color, by using the Color Picker tool, opening the color window and copying the last value. For this tutorial, I'll use purple, so the value I copied is 660066.



Second step - Converting the value to RGB values

Now, paste your copied value to this site: <http://corecoding.com/utilities/rgb-or-hex-to-float.php>
It will give you three values; one for Red, one for Green, and one for Blue. For the purple, I got:

R: 0.4

G: 0

B: 0.4

core coding®

RGB or HEX to Decimal Color Converter

Products Utilities Resources Links

→ RGB or HEX to Decimal Color Converter

Enter the HTML Color Code
such as E3B71A:

* OR *

Enter the 3 RGB Values:
Red:
Green:
Blue:

Copyright © 2003 - 2018 Core Coding®. Hosted by NDCHost. [About](#) [Contact](#) [FAQs](#) [Privacy Policy](#) [Terms of Service](#)

core coding®

RGB or HEX to Decimal Color Converter

Products Utilities Resources Links

→ RGB or HEX to Decimal Color Converter

Your decimal color codes are as follows:
0.4, 0, 0.4

Objective C UIColor Code:
[UIColor colorWithRed:0.4 green:0 blue:0.4 alpha:1.0];

[Start Over](#)

Copyright © 2003 - 2018 Core Coding®. Hosted by NDCHost. [About](#) [Contact](#) [FAQs](#) [Privacy Policy](#) [Terms of Service](#)

Third step - RGB values to hexadecimal

Now, head over to this site: <https://babbage.cs.qc.cuny.edu/IEEE-754.old/Decimal.html>

Copy only one of the RGB values at a time, and paste them on "Decimal Floating-Point", then click "Not Rounded" to generate the new values. Under Single precision (32 bits) you'll get a hexadecimal value. Do this with the three RGB values, and you'll get your *new* RGB values. For the purple, I got:

R: 3ECCCCC

G: 0000000

B: 3ECCCCC

IEEE-754 Floating-Point Conversion
From Decimal Floating-Point
To 32-bit and 64-bit Hexadecimal Representations
Along with Their Binary Equivalents

Enter a decimal floating-point number here,
then click either the **Rounded** or the **Not Rounded** button.

Decimal Floating-Point:

Rounding from floating-point to 32-bit representation uses the IEEE-754 round-to-nearest-value mode.

Results:

Decimal Value Entered:

Single precision (32 bits):

Binary: Status:

Bit 31 Sign Bit <input type="text"/> 0: + 1: -	Bits 30 - 23 Exponent Field <input type="text"/> Decimal value of exponent field and exponent <input type="text"/> - 127 = <input type="text"/>	Bits 22 - 0 Significand <input type="text"/> Decimal value of the significand <input type="text"/>
------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------

Hexadecimal: Decimal:

Double precision (64 bits):

Binary: Status:

Bit 63 Sign Bit <input type="text"/> 0: + 1: -	Bits 62 - 52 Exponent Field <input type="text"/> Decimal value of exponent field and exponent <input type="text"/> - 1023 = <input type="text"/>	Bits 51 - 0 Significand <input type="text"/> Decimal value of the significand <input type="text"/>
------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------

IEEE-754 Floating-Point Conversion
From Decimal Floating-Point
To 32-bit and 64-bit Hexadecimal Representations
Along with Their Binary Equivalents

Enter a decimal floating-point number here,
then click either the **Rounded** or the **Not Rounded** button.

Decimal Floating-Point:

Rounding from floating-point to 32-bit representation uses the IEEE-754 round-to-nearest-value mode.

Results:

Decimal Value Entered:

Single precision (32 bits):

Binary: Status:

Bit 31 Sign Bit <input type="text" value="0"/> 0: + 1: -	Bits 30 - 23 Exponent Field <input type="text" value="01111101"/> Decimal value of exponent field and exponent <input type="text" value="125"/> - 127 = <input type="text" value="-2"/>	Bits 22 - 0 Significand <input type="text" value="1.10011001100110011001100"/> Decimal value of the significand <input type="text" value="1.5999999"/>
----------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------

Hexadecimal: Decimal:

Double precision (64 bits):

Binary: Status:

Bit 63 Sign Bit <input type="text" value="0"/> 0: + 1: -	Bits 62 - 52 Exponent Field <input type="text" value="0111111101"/> Decimal value of exponent field and exponent <input type="text" value="1021"/> - 1023 = <input type="text" value="-2"/>	Bits 51 - 0 Significand <input type="text" value="1.100110011001100110011001100110011001100110011001"/> Decimal value of the significand <input type="text" value="1.600000000000000000000000"/>
----------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Fourth step - Hex editing a file

I want to edit Dragunov's 2P beret color, so I'll head over to:

- TekkenGame\Content\Character\Item\Customize\DRA\HEAD -

I then copy the file I want to edit, which is "CCI_dra_hed_2p", and then open it on the hex editor (If you want to edit a 1P costume color, you need to edit the file that has "_cus" in it's name). Once in hex, we'll need to find a material to guide ourselves onto what to edit. For example, on the text I see "MI CH dra hed 1P beret", so I know that any values that I change, correspond to that material.

[illegible]

A few rows under the material, you'll notice strings with 16 values that end in 00 00 80 3F (HxD will indicate that they have a length value of 10 at the bottom sidebar). These are the values that you want to change.

So what you need to do here, is replace the first 12 values with your hexadecimal RGB values from the last step, but there's a trick to it. For example, the R value I got is 3E CC CC CC, but I need to input it as CC CC CC 3E: so backwards, but not mirrored.

So the original 08 3D 0B 3E 53 23 E4 3E F6 28 3C 3F 00 00 80 3F value, ends up as CC CC CC 3E 00 00 00 00 CC CC CC 3E 00 00 80 3F.

Now save the file, and make a folder directory that's the same as the directory of the file you edited. So for me it would be "TekkenGame\Content\Character\Item\Customize\DRA\HEAD", and I put my edited file on that folder.

It's done! Your edited item now has a custom color; just pack the mod together by using u4pak and give it a try!

Our purple beret is done! Now, what shall *you* edit?

