Colors in Snap*!*

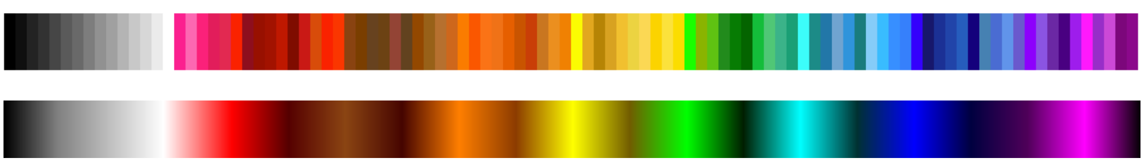
-bh proposal, draft, do not distribute

Your computer monitor can display millions of colors, but you probably can't distinguish that many. For example, here's red 57, green 180, blue 200: And here's red 57, green 182, blue 200: You might be able to tell them apart if you see them side by side: … but maybe not even then.

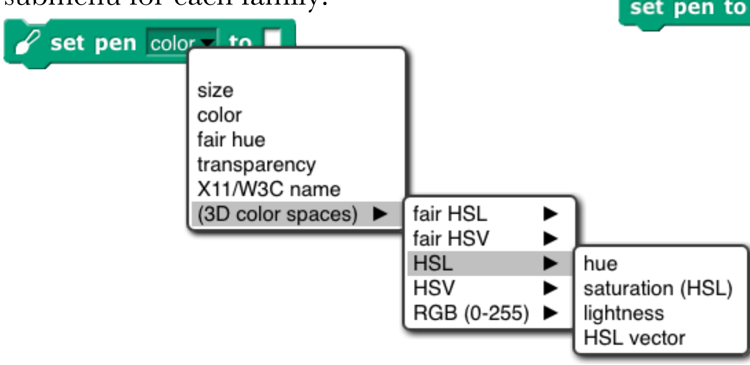
Color space—the collection of all possible colors—is three-dimensional, but there are many ways to choose the dimensions. RGB (red-green-blue), the one most commonly used, matches the way TVs and displays produce color. Behind every dot on the screen are three tiny lights: a red one, a green one, and a blue one. But if you want to print colors on paper, your printer probably uses a different set of three colors: CMY (cyan-magenta-yellow). You may have seen the abbreviation CMYK, which represents the common technique of adding black ink to the collection. (Mixing cyan, magenta, and yellow in equal amounts is supposed to result in black ink, but typically it comes out a not-very-intense gray instead.) Other systems that try to mimic human perception are HSL (hue-saturation-lightness) and HSV (hue-saturation-value).

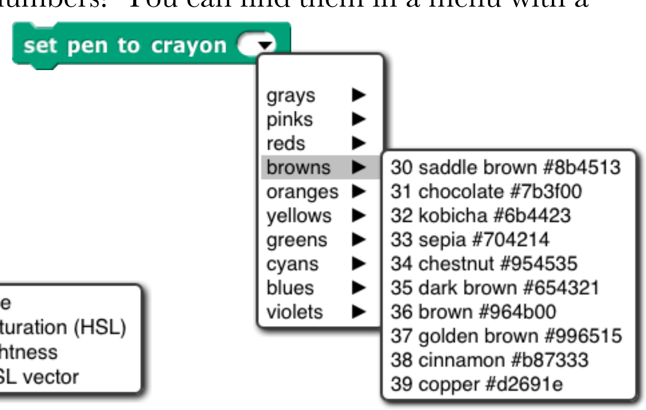
If you are a color professional—a printer, a web designer, a graphic designer, an artist—then you need to understand all this. It can also be interesting to learn about. For example, there are colors that you can see but your computer display can't generate. If that intrigues you, look up color theory in Wikipedia.

Crayons and Colors



But if you just want some colors in your project, we provide a simple, one-dimensional subset of the available colors. Two subsets, actually: *crayons* and *colors*. Here's the difference:
The first row shows 100 distinct colors. They have names; this is pumpkin, and this is denim. You're supposed to think of them as a big box of 100 crayons. They're arranged in families: grays, pinks, reds, browns, oranges, etc. But they're not ordered within a family, you'd be unlikely to say "next crayon" in a project. (But look at the crayon spiral on page 5.) Instead, you'd think "I want this to look like a really old-fashioned photo" and so you'd find sepia as crayon number 33. You don't have to memorize the numbers! You can find them in a menu with a submenu for each family.

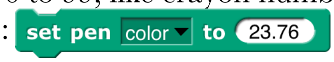




The crayon numbers are chosen so that skipping by 10 gives a sensible “box” of ten crayons:

Alternatively, skipping by 5 gives a still-sensible set of twenty crayons:

The Snap*!* block colors (Motion blue, Looks purple, etc.) are included among the crayons.

The set of *colors* is arranged so that each color is visually near each of its neighbors. Bright and dark colors alternate for each family. Color numbers range from 0 to 99, like crayon numbers, but you can use fractional numbers to get as tiny a step as you like: 

(“As tiny as you like” isn’t *quite* true because in the end, your color has to be rounded to integer RGB values for display.)

That’s all you have to know about colors! *Crayons* for specific interesting ones, *colors* for gradual transformation from one color to the next. But there’s a bit more to say, if you’re interested. If not, stop here. (But look at the samples of the different scales on page 5.)

Both of these scales include the range of shades of gray, from black to white. Since black is the initial pen color, and black isn’t a hue, Scratch and Snap*!* users would try to use SET COLOR to escape from black, and it wouldn’t work. By including black in the same scale as other colors, we eliminate the Black Hole problem if people use only the recommended color scales.

More about Colors: Fair Hues and Shades

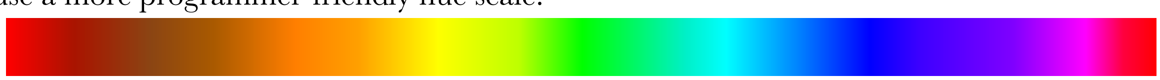
Several of the three-dimensional arrangements of colors use the concept of “hue,” which more or less means where a color would appear in a rainbow (magenta, at the right, is a long story):

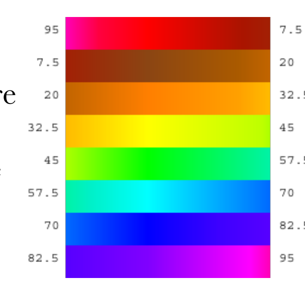
These are called “spectral” colors, after the “spectrum” of rainbow colors. But these colors aren’t equally distributed. There’s an awful lot of green, hardly any yellow, and just a sliver of orange. And there’s no brown at all.

And this is already a handwave, because the range of colors that can be generated by RGB monitors doesn’t include the *true* spectral colors. See Spectral color in Wikipedia for all the gory details.

This isn’t a problem with the physics of rainbows. It’s in the human eye and the human brain that certain ranges of frequency of light waves are lumped together as named colors. The eye is just “tuned” to recognize a wide range of colors as green. (See Rods and Cones.) And different human cultures give names to different color ranges. Nevertheless, in old Scratch projects, you’d say `change pen color by 1` and it’d take forever to reach a color that wasn’t green.

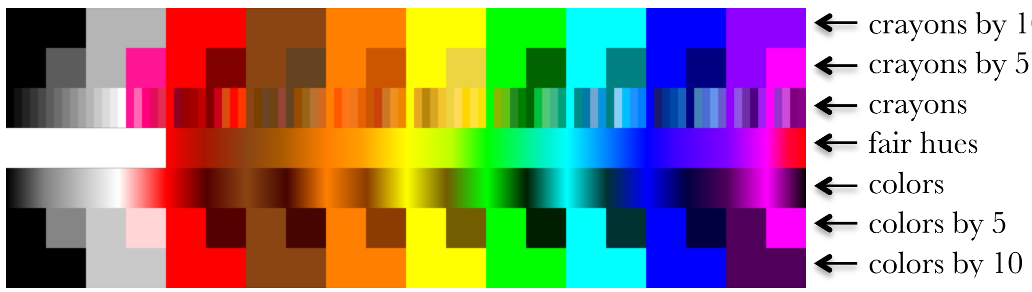
For color professionals, there are good reasons to want to work with the physical rainbow hue layout. But for amateurs using a simplified, one-dimensional color model, there’s no reason not to use a more programmer-friendly hue scale:





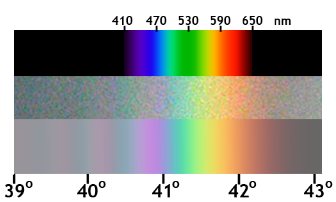
In this scale, each of the seven rainbow colors and brown get an equal share. (Red’s looks too small, but that’s because it’s split between the two ends: hue 0 is pure red, brownish reds are to its right, and purplish reds are wrapped around to the right end.) We call this scale “fair hue” because each color family gets a fair share of the total hue range. (By the way, you were probably taught “… green, blue, indigo, violet” in school, but it turns out that color names were different in Isaac Newton’s day, and the color he called “blue” is more like modern cyan, while his “indigo” is more like modern blue. See Wikipedia Indigo.)

Our *color* scale is based on fair hues, adding a range of grays from black (color 0) to white (color 14) and also adding *shades* of the hue colors. (In color terminology, a *shade* is a darker version of a color; a lighter version is called a *tint*.) Why do we add shades but not tints? Partly because I find shades more exciting. A shade of red can be dark candy apple red or maroon, but a tint is just some kind of pink. This admitted prejudice is supported by an objective fact: Most projects are made on a white background, so dark colors stand out better than light ones. So, in our color scale, colors 0 to 14 are kinds of gray; the remaining colors go through the fair hues, but alternating full-strength colors with shades.



This chart shows how the color scales discussed so far are related. Note that all scales range from 0 to 100; the fair hues scale has been compressed in the chart so that similar colors line up vertically. (Its dimensions are different because it doesn't include the grays at the left. Since there are eight color families, the pure, named spectral colors are at multiples of 100/8=12.5, starting with red=0.)

White is crayon 14 and color 14. This value was deliberately chosen *not* to be a multiple of 5 so that the every-fifth-crayon and every-tenth-crayon selections don't include it, so that all of the crayons in those smaller boxes are visible against a white stage background.



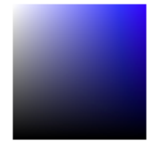
Among purples, the official spectral violet (crayon and color 90) is the end of the spectrum. Magenta, brighter than violet, isn't a pure spectral color at all. (In the picture at the left, the top part is the spectrum of white light spread out through a prism; the middle part is a photograph of a rainbow, and the bottom part is a digital simulation of a rainbow.) Magenta is a mixture of red and blue.
(attribution: Wikipedia user Andys. CC BY-SA.)

The light gray at color 10 is slightly different from crayon 10 just because of roundoff in computing crayon values. Color 90 is different from crayon 90 because in the purple family the darker color comes first; color and crayon 95 are the same. Otherwise, the colors, crayons, and (scaled) fair hues all agree at multiples of ten. These multiple-of-ten positions are the standard RGB primary and secondary colors, e.g., the yellow at color 50 is (255, 255, 0) in RGB. (Gray, brown, and orange don't have such well-defined RGB settings.)

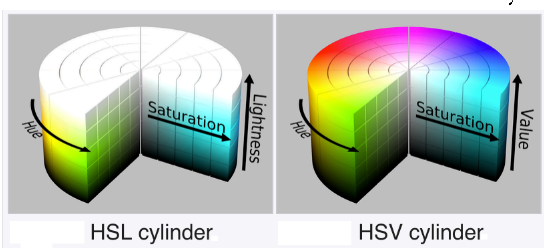
The colors at odd multiples of five are generally darker shades than the corresponding crayons. The latter are often official named shades, e.g., teal, crayon 65, is a half-intensity shade of cyan. The odd-five colors, though, are often darker, since the usable color range in a given family has this as its darkest representative. The pink at color 15, though, is quite different from crayon 15, because the former is a pure tint of red, whereas the crayon, to get a more interesting pink, has a little magenta mixed in. Colors at multiples of five are looked up in a table; other color values are determined by linear interpolation in RGB space. (*Crayons* are of course all found by table lookup.)

Perceptual Spaces: HSL and HSV

RGB is the right way to think about colors if you're building or programming a display monitor; CMYK is the right way if you're building or programming a color printer. But neither of those coordinate systems is very intuitive if you're trying to understand what color *you see* if, for example, you mix 37% red light, 52% green, and 11% blue. The *hue* scale is one dimension of most attempts at a perceptual scale. The square at the right has pale blues along the top edge, dark blues along the right edge, various shades of gray toward the left, black at the bottom, and pure spectral blue in the top right corner. Although no other point in the square is pure blue, you can tell at a glance that no other spectral color is mixed with the blue.



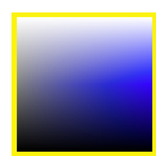
Aside from hue, the other two dimensions of a color space have to represent how much white and/or black is mixed with the spectral color. (Bear in mind that "mixing black" is a metaphor when it comes to monitors. There really is black paint, but there's no such thing as black light.)



attribution: Wikipedia user SharkD, CC BY-SA 3.0

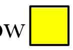
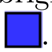
One such space, HSV, has one dimension for the amount of color (vs. white), called *saturation*, and one for the amount of black, imaginatively called *value*. HSV stands for Hue-Saturation-Value. (I don't know why they couldn't think of a more descriptive name.) The *value* is actually measured backward from this description; that is, if value is 0, the color is pure black; if value is 100, then a saturation of 0 means all white, no spectral color; a saturation of 100 means no white at all. In the square in the previous paragraph, the *x* axis is the saturation and the *y* axis is the value. The entire bottom edge is black, but only the top left corner is white. HSV is the traditional color space used in Scratch and Snap*!*. `Set pen color` set the hue; `set pen shade` set the value. There was originally no Pen block to set the saturation, but there's a `set brightness effect` Looks block to control the saturation of the sprite's costume. (I speculate that the Scratch designers, like me, thought tints were less vivid than shades against a white background.)

But if you're looking at colors on a computer display, HSV isn't really a good match for human perception. Intuitively, black and white should be treated symmetrically. This is the HSL (hue-saturation-lightness) color space. *Saturation* is a measure of the *grayness* or *dullness* of a color (how close it comes to being on a black-and-white scale) and *lightness* measures *spectralness* with pure white at one end, pure black at the other end, and full spectral color in the middle. The *saturation* number is actually the opposite of grayness: 0 means pure gray, and 100 means pure spectral color, provided that the *lightness* is 50, midway between black and white. Colors with lightness other than 50 have some black and/or white mixed in, but saturation 100 means that the color is as fully saturated as it can be, given the amount of white or black needed to achieve that lightness. Saturation less than 100 means that *both white and black* are mixed with the spectral color. (Such mixtures are called *tones* of the spectral color. Perceptually, colors with saturation 100% don't look gray:  but colors with saturation 75% do: 



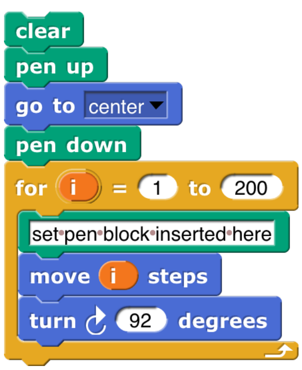
Note that HSV and HSL both have a dimension called "saturation," but *they're not the same thing!* In HSV, "saturation" means non-whiteness, whereas in HSL it means non-grayness (vividness).

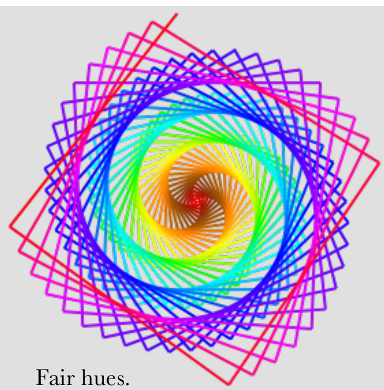
Although traditional Scratch and Snap*!* use HSV in programs, they use HSL in the color picker. The horizontal axis is hue (fair hue, in this version) and the vertical axis is *lightness,* the scale with black at one end and white at the other end. It would make no sense to have only the bottom half of this selector (HSV Value) or only the top half (HSV Saturation). And, given that you can only fit two dimensions on a flat screen, it makes sense to pick HSL saturation (vividness) as the one to keep at 100%. (In this fair-hue picker, some colors appear twice: "spectral" (50% lightness) browns as shades (≈33% lightness) of red or orange, and shades of those browns.)

Software that isn't primarily about colors (so, *not* including Photoshop, for example) typically use HSV or HSL, with web-based software more likely to use HSV because that's what's built into the JavaScript programming language provided by browsers. But if the goal is to model human color perception, neither of these color spaces is satisfactory, because they assume that all full-intensity spectral colors are equally bright. But if you're like most people, you see spectral yellow  as much brighter than spectral blue . There are better perceptual color spaces with names like L*u*v* and L*a*b* that are based on research with human subjects to determine true visual brightness. Wikipedia explains all this and more at [HSL and HSV](), where they recommend ditching both of these simplistic color spaces. ☺

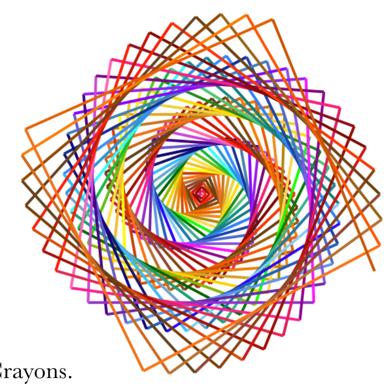
tl;dr

For normal people, Snap*!* provides three simple, one-dimensional scales: *crayons* for specific interesting colors, *colors* for a continuum of high-contrast colors with a range of hues and shading, and *fair hues* for a continuum without shading. For color nerds, it provides three-dimensional color spaces RGB, HSL, HSV, and fair-hue variants of the latter two. We recommend "fair HSL" for zeroing in on a desired color.

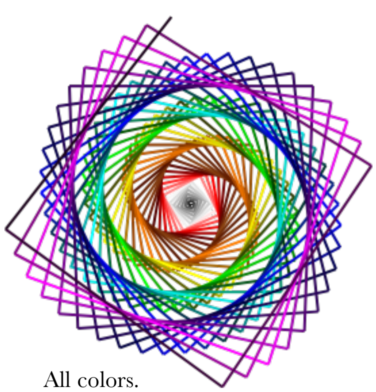




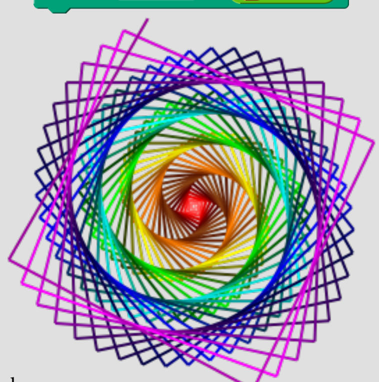
Fair hues.



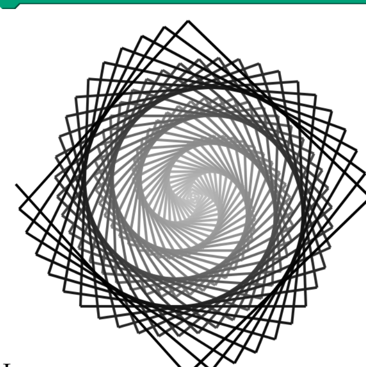
Crayons.



All colors.

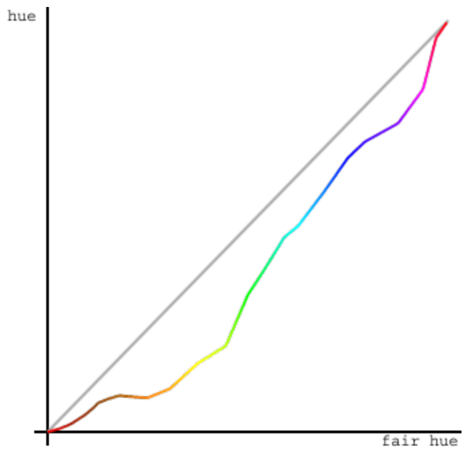


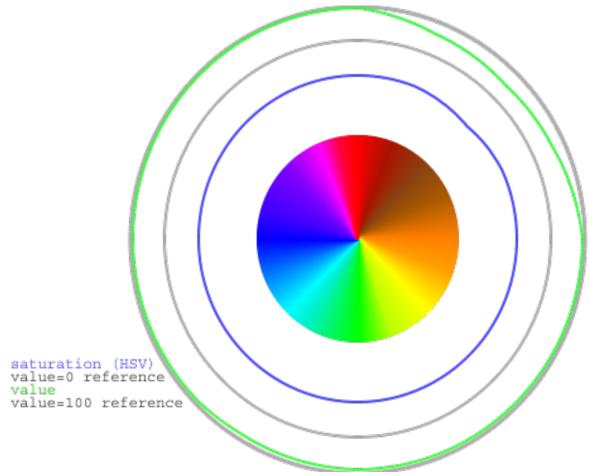
Colors, no grays.



Just grays.

Appendix: Geeky details on fair hue



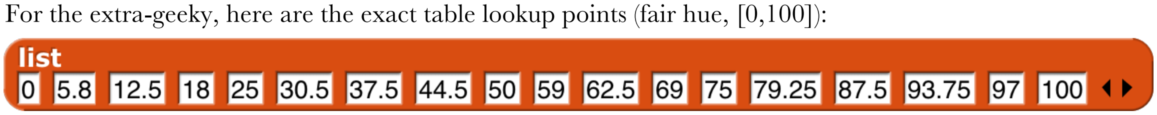


The left graph shows that, unsurprisingly, all of the brown fair hues make essentially no progress in real hue, with the orange-brown section actually a little retrograde, since browns are really shades of orange and so the real hues overlap between fair browns and fair oranges. Green makes up some of the distance, because there are too many green real hues and part of the goal of the fair hue scale is to squeeze that part of the hue spectrum. But much of the catching up happens very quickly, between pure magenta at fair hue 93.75 and the start of the purple-red section at fair hue 97. This abrupt change is unfortunate, but the alternatives involve either stealing space from red or stealing space from purple (which has to include both spectral violet and RGB magenta). The graph has discontinuous derivative at the table-lookup points, of which there are two in each color range, one at the pure-named-RGB colors at multiples of 12.5, and the other *roughly* halfway to the next color range, except for the purple range, which has lookup points at 87.5 (approximate spectral violet), 93.75 (RGB magenta), and 97 (turning point toward the red family). (In the color picker, blue captures cyan and purple space in dark shades. This, too, is an artifact of human vision.)

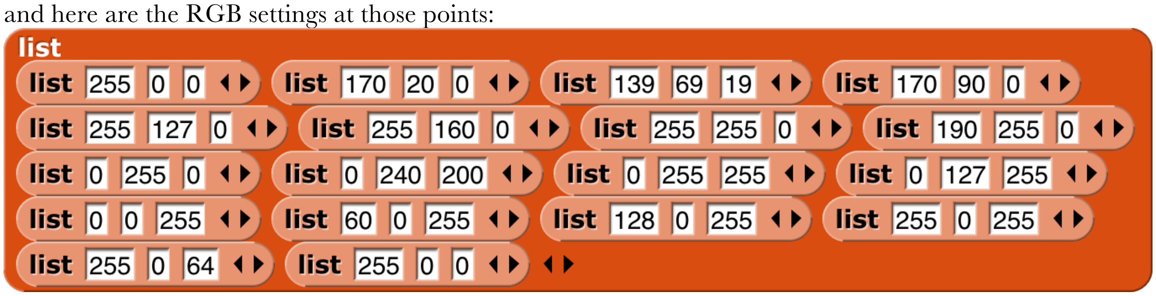
The right graph shows the HSV saturation and value for all the fair hues. Saturation is at 100%, as it should be in a hue scale, except for a very slight drop in part of the browns. (Browns are shades of orange, not tints, so one would expect full saturation, except that some of the browns are actually mixtures with related hues.) But value, also as expected, falls substantially in the browns, to a low of about 56% (halfway to black) for the "pure" brown at 45° (fair hue 12.5). But the curve is smooth, without inflection points other than that minimum-value pure brown.

"Fair saturation" and "fair value" are by definition 100% for the entire range of fair hues. This means that in the browns, the real saturation and value are the product (in percent) of the innate shading of the specific brown fair hue and the user's fair saturation/value setting. When the user's previous color setting was in a real scale and the new setting is in a fair scale, the program assumes that the previous saturation and value were entirely user-determined; when the previous color setting was in a brown fair hue and the new setting is also in a fair scale, the program remembers the user's intention from the previous setting. (Internal calculations are based on HSV, even though we recommend HSL to users, because HSV comes to us directly from the JavaScript color management implementation.) This is why the `set pen` block includes options for "fair saturation" and so on.

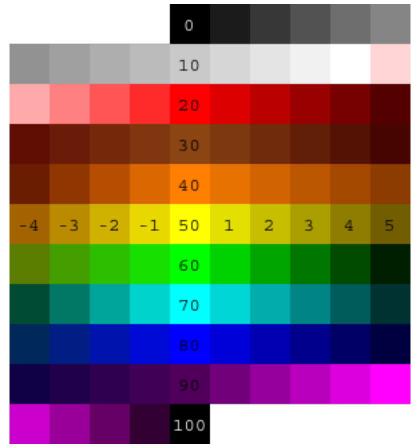
For the extra-geeky, here are the exact table lookup points (fair hue, [0,100]):



and here are the RGB settings at those points:



Appendix: Geeky details on colors



Here is a picture of integer color numbers, but remember that color numbers are continuous. (As usual, "continuous" values are ultimately converted to integer RGB colors, so there's really some granularity.) Colors 0-14 are continuously varying grayscale, from 0=black to 14=white. Colors 14+ε to 20 are linearly varying shades of pink, with RGB Red at color 20.

Beyond that point, in each color family, the multiple of ten color number in the middle is the RGB standard color of that family, in which each component is either 255 or 0. (Exceptions are brown, which is of course darker than any of those colors; orange, with its green component half-strength: [255, 127, 0]; and violet, discussed below.) The following multiple of five is the darkest color in that family, although not necessarily the same hue as the multiple of ten color. Colors between the multiple of ten and the following multiple of five are shades of colors entirely within the family. Colors in the four *before* the multiple of ten are mixtures of this family and the one before it. So, for example, in the green family, we have

55	Darkest yellow.
(55, 60)	shades of yellow-green mixtures. As the color number increases, both the hue and the lightness (or value, depending on your religion) increase, so we get brighter and greener colors.
60	Canonical green, [0, 255, 0], whose W3C color name is "lime," not "green."
(60, 65)	Shades of green. No cyan mixed in.
65	Darkest green.
(65,70)	Shades of green-cyan mixtures.

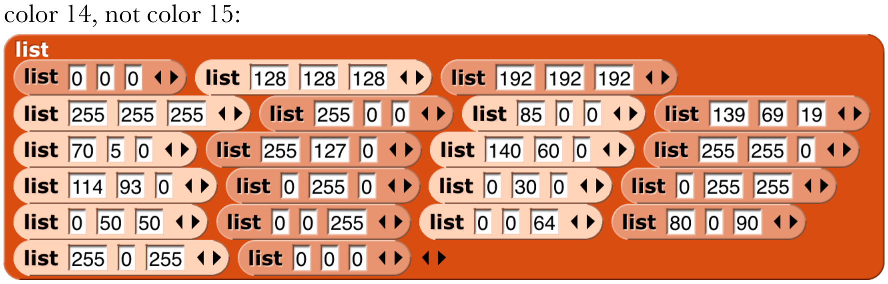
In the color chart, all the dark colors look a lot like black, but they're quite different. Here are the darkest colors in each family.

Darkest yellow doesn't look entirely yellow. You might see it as greenish or brownish. As it turns out, the darkest color that really looks yellow is hardly dark at all. This color was hand-tweaked to look neither green nor brown to me, but ymmv.

In some families, the center+5 *crayon* is an important named darker version of the center color: In the red family, [0, 128, 0] is "maroon." In the cyan family, [0, 128, 128] is "teal." An early version of the color scale used these named shades as the center+5 color also. But on this page we use the word "darkest" advisedly: You can't find a darker shade of this family anywhere in the color scale, but you *can* find lighter shades. Teal is color number 73.1, ($70 + 5 \cdot \frac{255-128}{255-50}$), because darkest cyan, color 75, is [0, 50, 50]. The color number for maroon is left as an exercise for the reader.

The purple family is different from the others, because it has to include both spectral violet and extraspectral RGB magenta. Violet is usually given as [128, 0, 255], but that's much brighter than the violet in an actual spectrum (see page 3). We use [80, 0, 90], a value hand-tweaked to look as much as possible like the violet in rainbow photos, as color 90. (*Crayon* 90 is [128, 0, 255].) Magenta, [255, 0, 255], is color 95. This means that the colors get *brighter,* not darker, between 90 and 95. The darkest violet is actually color 87.5, so it's bluer than standard violet, but still plainly a purple and not a blue. It's [39,0,76]. It's *not* hand-tweaked; it's a linear interpolation between darkest blue, [0, 0, 64], and the violet at color 90. I determined by experiment that color 87.5 is the darkest one that's still unambiguously purple. (According to Wikipedia, "violet" names only the spectral color, while "purple" is the name of the whole color family.)

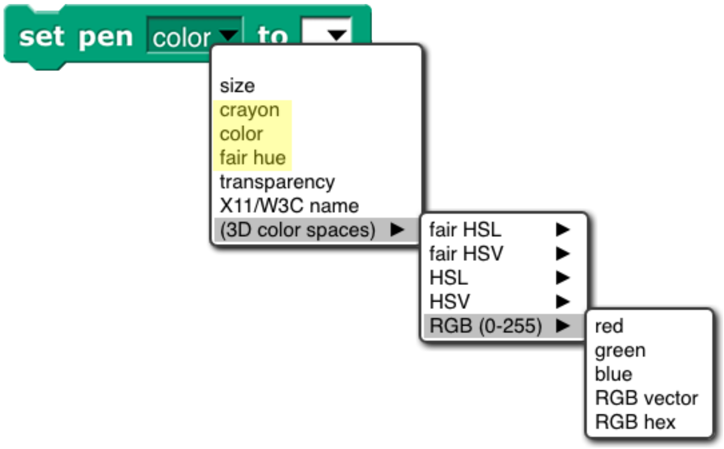
Here are the color reference points, for color numbers that are multiples of five, except for item 4, which is used for color 14, not color 15:



The very pale three-input list blocks are for color numbers that are odd multiples of five, generally the "darkest" members of each color family. (The block colors were adjusted in Photoshop; don't ask how to get blocks this color in Snap*!*.)

Appendix: For the Snap*!* Reference Manual

The proposal is to merge the **set pen** and **set pen to crayon** blocks. The combined block will look like this; the three menu items highlighted here are the three linear color scales intended for non-color-nerd users.



The 3-D color space options are in submenus. Each of the latter includes a "vector" menu item; if a vector option is chosen, then the second input is expected to be a list of three or four numbers. The first three are interpreted according to the rules of the chosen color space; in particular, the effective range of numbers is [0,100] for hues, [0,100] for saturation, lightness, or value. RGB values are [0, 255]. The fourth number, if present, is transparency in percent. (0 is opaque; 100 is invisible.)

An RGB color can also be specified as a hexadecimal numeral. If that numeral is exactly three digits long, each digit is multiplied by 17, so that `234` becomes `223344`. Otherwise the numeral is extended with zeros on the left to make six hexadecimal digits.

Another representation for colors is the collection of 650-ish official [X11/W3C color names](), such as "antique white" and "powder blue." (Some, but not all, of the 100 crayon names are taken from this list.) If the user-supplied name is not an exact match for one of the names in the list, the **set pen** block looks first for any partial match; if more than one is found, the search is limited to leading partial matches (e.g., "blue" wouldn't match "powder blue" but would match "blue violet." If there are still multiple matches, the search is tried once more, omitting trailing digits. (There are several families of four numbered colors.) These searches are tried in the order just given, so in fact it doesn't matter about "blue" as a partial match to longer names, because there is a color whose entire name is "blue" (unsurprisingly, `0000ff`) and that is found right away. By contrast, "green" is not resolved by any of these trials, because green is one of the four color names that represent different colors in the X11 list (`00ff00`) and the W3C list (`008000`), so the block throws an error message telling the user to choose "green (x11)," "green (w3c)," or "greenyellow." (There's a much larger Wikipedia color list, a superset of this one, with entries such as "caput mortuum" (`592720`), "dark liver (horses)" (`543d37`), and "fuzzy wuzzy" (`87421f`). Someday.) There's nothing like regular expression matching; the idea is that the user already knows what color name they want, not that they want a list of the 26 color names with "blue" as a substring.

The second input slot offers different menus depending on the first input. In particular, this block incorporates the crayon menu organized by family. For other scales, the second-slot menu may include only some named values, much as the **point in direction** block allows any number but also has names for the four horizontal and vertical cases. Hue scales will have names for the standard RGB rainbow names (plus brown, for fair hue). The color scale's menu will include those plus the "darkest X" values and a few common named colors such as teal and maroon. Most options, though, don't offer menus in the second slot.

The **pen** reporter, whose one input slot has the same menu as the first input to **set pen**, can always report RGB, HSL, or HSV values for the current pen color. But color, fair hue, crayon, and X11/W3C values can be reported only if the current pen color was set in the corresponding way. The **change pen** block works only for numeric scales. ("RGB hex" is not a numeric scale; it encodes three numeric dimensions, not one.)

Read the algorithms or try out the blocks at
https://snap.berkeley.edu/snapsource/snap.html#present:Username=bh&ProjectName=super%20set%20pen