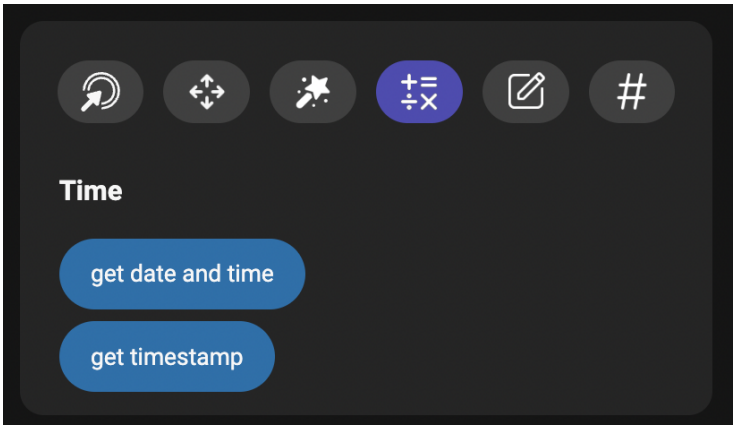
Horizon Worlds: System Time Code Blocks

An overview of how to use system time code blocks.

Definition of code blocks

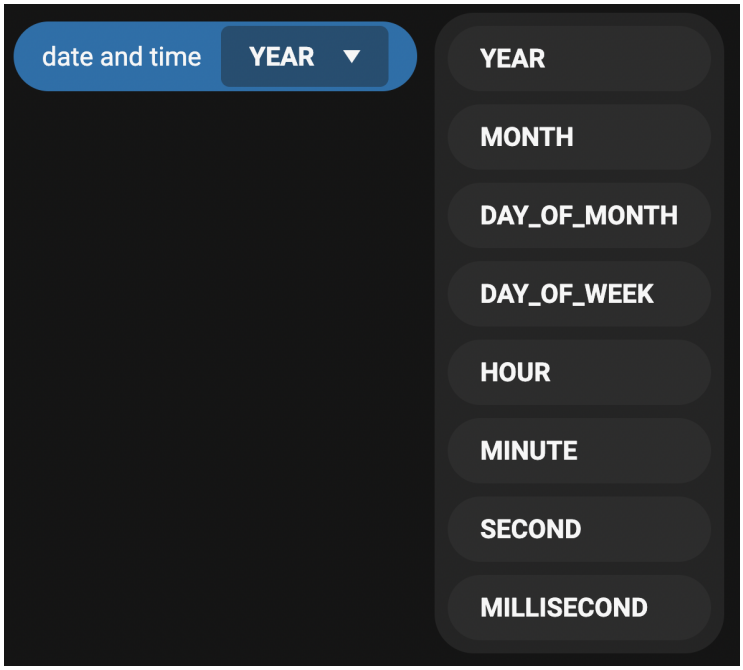
Two code blocks can be found under **Operators > Time**



Get Date and Time

Gets the current date and time in the UTC timezone and returns a number for the requested “component”:

- Milliseconds (a integer value 0 to 999)
- Seconds (an integer 0 to 59)
- Minute (an integer 0 to 59)
- Hour (an integer 0 to 23; 0 = midnight, 12 = noon, etc)
- Day of the week (an integer 0 to 6; 0 = Sun, 1 = Mon, etc)
- Day of the month (an integer 1 to 31)
- Month (an integer 1 to 12)
- Year (an integer)

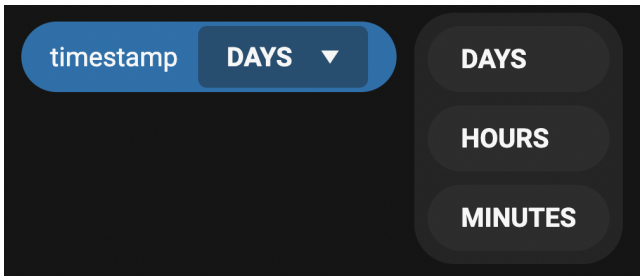


You can use these values in different combinations to work with the full date and time (at this exact moment in the UTC timezone). See example #1.

Get Timestamp

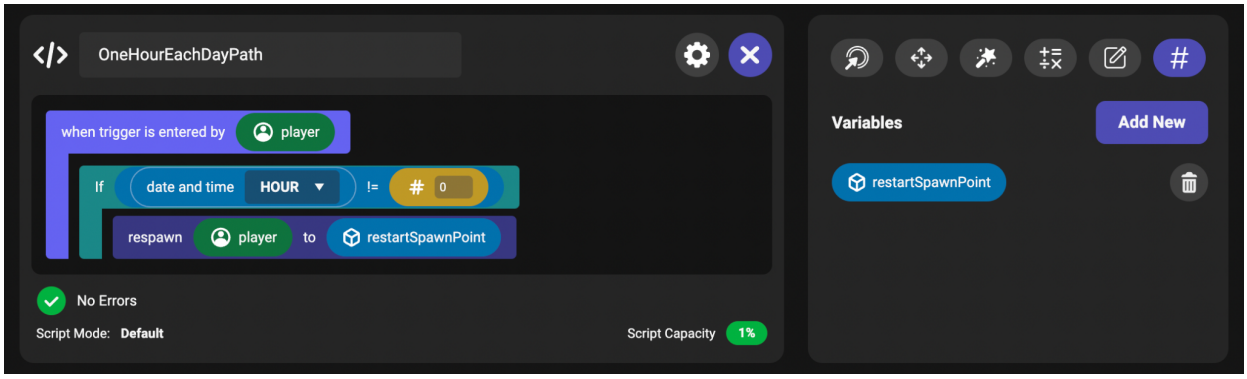
Returns a number equal to the amount of time that has passed since 12:00a Jan 1, 2022 based on what the user picks in the drop down. Note these numbers are totals since the start time, so “minutes” will be a big number, for example. The options are:

- Minutes: a decimal number
- Hours: a decimal number
- Days: a decimal number



Generally you won’t use these numbers individually but instead will compare two numbers (e.g. by subtraction). E.g. if you store the “minutes” in a persistent variable when a person leaves a world, then the next time they arrive, you can figure out how long it has been since they were last here. See example #2.

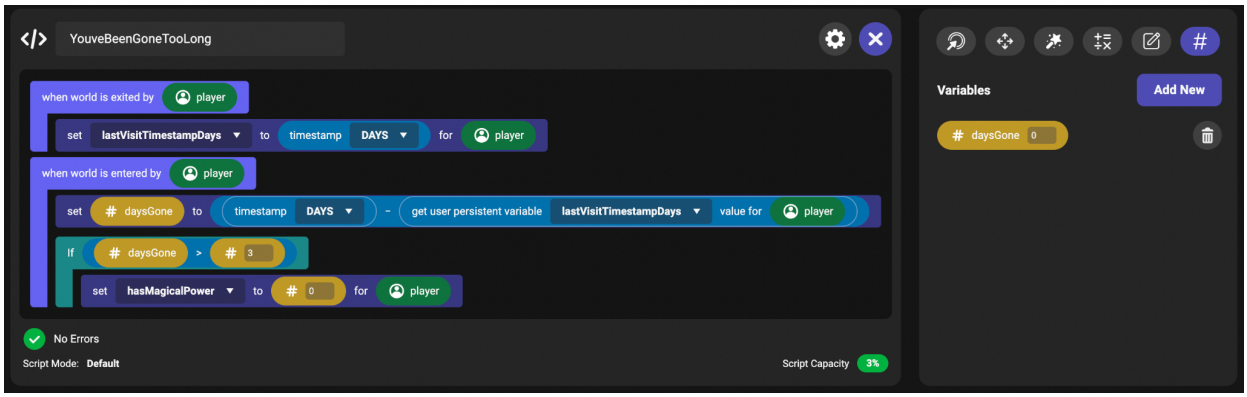
Example 1 - a tunnel open for 1 hour a day



What it does: in this example there is a tunnel that you can only walk through between 12:00 am and 12:59:59 am UTC (which is 5-6PM in Pacific Time, e.g. California). If you walk into the tunnel outside of that time, it respawns you back to a spawn point at the start.

How it works: when a player enters the trigger, we get the current hour in the UTC timezone. If the hour is not 0 (meaning we are not in the time range 0:00 to 0:59:59) then we use the respawn code block to send the player back.

Example 2 - lose a power up if gone too long



What it does: in this example whenever a player enters the world, if it has been more than 3 days since they were last there then they “lose” their magical powers. You need to visit the world every few days or you will lose your powers!

How it works: when a player leaves the world we set a persistent variable of current timestamp in minutes. The next time they come to the world we get a new timestamp. If the new timestamp is more than 3 days since you were last there, we clear the persistent variable tracking whether or not you have magical powers. Sorry, you should have come again sooner! ;)

Some Use Case Ideas

- Parts of the world only unlock at certain times of the day or week

- “Visit streak” if you visit every day you increasingly unlock items; break your streak and lose them
- Change the sky through the day with asset spawning to change the environment gizmo over time

Warning - Current Limitation

Right now, every time you get the date / time or timestamp, it gets the exact time at that moment. That means that when calling the code blocks back to back, you may run into issues. For example, imagine that right now it is 3.99999 seconds into the current minute and then you get the current number of seconds. The code block will return **3**, then when you call milliseconds you might get **999** or you might get **0**, since a tiny amount of time has passed since the previous code block.

This means that if you get seconds and then milliseconds back-to-back and get 3 and then 0, this is a tiny chance that it should have been 4 and 0 at that point. Right now this is no easier work around for this case. Be mindful of getting multiple time components in a row, since you will get this “wraparound problem”.

FAQ

Can these code blocks be called from local scripts?
You *can* but be aware that people can change the time on their headsets so it is recommended to only call these blocks in *default* scripts.

If I get the current “milliseconds” two frames in a row and subtract them, can I use that as the frame “delta time”?
Not realistically. Milliseconds are returned as an integer, and the same code might run at slightly different times across two frames (e.g. early in one frame and late in another frame). Thus you are likely to incur significant error when trying to calculate the time between two frames via these code blocks.

Additionally, you’d need to use seconds and milliseconds together to address how these numbers wrap back around (modulo), a calculation that is easy to get wrong.

Are these blocks affected by daylight savings time?
No. UTC does not have daylight savings time.

Are subsequent calls to these codeblocks guaranteed to be increasing?
When *called from the server* these codeblocks will always increase (except for the crazy-rare leap second where a second will repeat twice). Calling these from local scripts at the moment can cause time discrepancies between different owners and people can change the times on

their headsets (causes the time to jump forward or back). Thus it is recommended to only call these blocks in *default* scripts.

How are these code blocks affected by leap seconds?
There hasn't been a leap second since December 2016. There have only ever been 27 of them total (as of June 2022) and they always take place on the last day of June or December. When, or if, a leap second occurs the same "second" value will be reported twice for two consecutive seconds.