📄 **Instructions**   </> Code Examples   ▶ Screencasts   💬 Forum   **Stage #BF7**   (**Pending**)

Write   Preview

Found an interesting resource? Share it with the community.

[M↓] Markdown supported.   Comment

👤 **sarp** challenge author   9 months ago

pieces field is of type *string*, but its content is actually *binary data*. when you open the torrent file with a text editor, you'll see weird characters, because in a text editor, each byte is shown o screen as a character (remember the ASCII table) and not all values correspond to readable characters

so if we treat this string as a sequence of bytes, the first 20 of them will be the first piece hash, next 20 will be the second piece hash etc. when we say byte, it's just a number between 0 and 255 (because we store it as an 8 bit unsigned integer)

so a piece hash could look like this as a sequence of bytes:
`[232, 118, 246, 122, 42, 136, 134, 232, 243, 107, 19, 103, 38, 195, 15, 162, 151, 3, 2, 45`

it's just 20 numbers. when we print each number in hexadecimal format, it will look like this:
`e876f67a2a8886e8f36b136726c30fa29703022d`

because `232` is represented as `0xe8` in hexadecimal (16^0 * 8 + 16 ^ 1 * 14) and we omit the 0x part
since we use 2 characters to print each byte, the printed piece hash will be 40 characters long

In Terminal, you can use `hexdump -C sample.torrent` to view the hex values and corresponding ASCII characters side by side

∧ **24**   ∨   ↰ reply

Ready to run tests...   [Show logs]