Ruby - Bug #18972

String#byteslice should return BINARY (aka ASCII-8BIT) Strings

08/23/2022 11:03 AM - byroot (Jean Boussier)

Status:	Rejected		
Priority:	Normal		
Assignee:			
Target version:			
ruby -v:		**Backport:**	2.7: UNKNOWN, 3.0: UNKNOWN, 3.1: UNKNOWN

Description

While working on implementing https://bugs.ruby-lang.org/issues/13626, I noticed byteslice assign the receiver encoding to the returned String.

I believe this is incorrect, as since you are doing a byte based operation, you do expect a binary string in return, otherwise if you'd call it on an UTF-8 string, you'd likely get a string with invalid encoding.

I read the original feature request and there's no mention of what the returned encoding should be:
https://bugs.ruby-lang.org/issues/4447

Current behavior

```
>> "fée".byteslice(1).valid_encoding?
=> false
>> "fée".byteslice(1).encoding
=> #<Encoding:UTF-8>
```

Expected behavior

```
>> "fée".byteslice(1).valid_encoding?
=> true
>> "fée".byteslice(1).encoding
=> #<Encoding:ASCII-8BIT>
```

Backward compatibility concerns

I'm honestly not quite sure what the backward incompatibility impact may be.

From my point of view if you are calling byteslice it's to use it with other binary string, but it's indeed
possible that there is existing code mixing UTF-8 and BINARY that somewhat work and would be broken by this change.

Especially since binary strings can silently be promoted from BINARY to UTF-8:

```
buffer = "".b
buffer << "fée" # buffer was promoted to Encoding::UTF-8 silently
buffer << "fée".byteslice(1)
```

The above currently "works", but would raise Encoding::CompatibilityError with this change.

Related issues:

Related to Ruby - Feature #4447: add String#byteslice() method	**Closed**	**02/25/2011**
Related to Ruby - Feature #13626: Add String#byteslice!	**Open**	

History

#1 - 08/23/2022 01:31 PM - byroot (Jean Boussier)

- Related to Feature #4447: add String#byteslice() method added

#2 - 08/23/2022 01:31 PM - byroot (Jean Boussier)

- Related to Feature #13626: Add String#byteslice! added

#3 - 08/23/2022 01:44 PM - Eregon (Benoit Daloze)

I think the current behavior is better, String#byteslice is not only used for BINARY strings.
In fact for binary strings (and other fixed-width encodings), there is no point to use byteslice over slice/[].

For instance, one might work with UTF-8 and get a byte index (instead of a character index), from e.g. String#byteindex or from MatchData#byteoffset, and then one would use byteslice to avoid 2 extra byte offset<->character offset conversions, which e.g. are expensive for (non-7-bit) UTF-8.
What I just described is close to the motivation for #13110 which added String#byteindex.

So I think we cannot change this for compatibility, and it is intended AFAIK.

#4 - 08/23/2022 03:49 PM - byroot (Jean Boussier)

- Status changed from Open to Rejected

Ok, I suppose your point of view make sense, and either way the backward compatibility concern is just too big.

Closing.