# 411.https://stackoverflow.com/questions/71181719/turning-bytes-into-string-for-concatenation

**T:**Turning bytes into string for concatenation

**Q:**I have this function in my smart contract:  
  
 function getBackgroundColor(uint256 tokenId) public pure returns (string memory) { uint256 rand = random(string(abi.encodePacked(toString(tokenId)))); bytes32 val = bytes32(rand); bytes memory hx = "0123456789ABCDEF"; bytes memory str = new bytes(26); for (uint i = 17; i < 20; i++) { str[i\*2] = hx[uint(uint8(val[i + 12] >> 4))]; str[1+i\*2] = hx[uint(uint8(val[i + 12] & 0x0f))]; } return string(str); }  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
I use it like so:  
  
function tokenURI(uint256 tokenId) override public pure returns (string memory) { string[17] memory parts; string memory backgroundColor = getBackgroundColor(tokenId); parts[0] = '<svg xmlns="http://www.w3.org/2000/svg" preserveAspectRatio="xMinYMin meet" viewBox="0 0 512 512"><rect width="100%" height="100%" fill="#'; parts[1] = backgroundColor; parts[2] = '"/><path d="M256 372.057L378 203.686L331.4 140H180.6L134 203.686L256 372.057Z" fill="hsl(';... string memory output = string(abi.encodePacked(parts[0], parts[1], parts[2], parts[3], parts[4], parts[5], parts[6], parts[7], parts[8])); output = string(abi.encodePacked(output, parts[9], parts[10], parts[11], parts[12], parts[13], parts[14], parts[15], parts[16])); string memory json = Base64.encode(bytes(string(abi.encodePacked('{"name": "Fweb3 NFT", "description": "This NFT represents participation in Fweb3.", "image": "data:image/svg+xml;base64,', Base64.encode(bytes(output)), '"}')))); output = string(abi.encodePacked('data:application/json;base64,', json)); return output;  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
Unfortunately it shows up like this in the rendered SVG:  
  
...<rect width="100%" height="100%" fill="#����������������������������������000001�����������">...  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
How would I include it into the resulting string more cleanly?

0 **Answer**