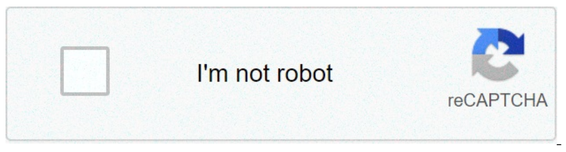

Continue

How many tranq arrows for trike

How many tranq arrows for a trike lvl 140. How many tranq arrows for a trike lvl 2. How many tranq arrows for a trike lvl 90. Ark survival evolved how many tranq arrows for trike. How many tranq arrows for a trike 145. How many tranq arrows for a trike lvl 50. How many tranq arrows for a trike lvl 100. How many tranq arrows for a trike lvl 15.

My Ark domestic calculator calculates food, time and narcotics for dominating the various creatures in the Ark. This page is to contain comments on the calculator and provide some background. This page was written by V190 of ARK. Updated from V234. The implementations of all the calculations discussed below can be found in the Taming calculator. The source code can be found here. Start now serve as Changelog. To-do: Do not allow invalid entries in all fields Add comibilites Determine the steps to make the code work when it is downloaded Add new dinos / ko methods / food dev kit DATA Many of the figures needed for the calculation of Taming are available via the Dev Kit. Instructions for acquiring the Dev kit are here. Once in the Dev kit, some useful numbers can be found by clicking on the 'PrimAleArth' folder, filtering through projects and entering the (inside) name of the Dino in the search bar, as seen in the image on the right. A list of internal names can be found in PrimAleArth-> Dinos, since every creature has a folder in that folder. Three models are selected in the image, these are the ones that are relevant to tame the calculation. The first is [creature] _character_bp. This project contains information about the dominant affinity of the creature. Filtering Blueprint properties for the term "AFF" will reduce the list to the relevant values. "Required abdominal affinity" is the basic level of affinity required for a creature of this type at level 0. "Required Affinity Affinity for Basic Level" is the gain of affinity for level of the creature (here "base level" presumably refers to the Before Taming level, as creatures now gain levels when they dominate). A level 1 creature requires initial affinity plus one level of affinity to the level. You'll also see references to domesticated inefficiency, which is supposed to be used for efficacy management calculation and "Waking Tames" which is the new non-violent tamme method. Next UP is DinoCharacTerstatusComponent_BP_ [Creature]. This draft contains details on the creature food consumption rates and torpidification rates. Filter by 'food' to get the food values. The value of the 'food' at the top seem to the creatures base amount of food – the total food is not of interest to tamper (starving is generally not a risk-tacking wealth) so I did nothing with it. The numbers we want are lower – "Basic Food Consumption Rate" and "Pron water food consumption multiplier." The first is exactly what it says it is – the food consumption rate of creatures. It says "base", but it doesn't seem to vary by level (at least not when taming). This is a very small value, presumably so that Dinos It lasts so long without food if they don't move. This last value is a multiplier who, despite the odd name, is applied when the Dino is unfolding and dominated. Muleply these two numbers to get the actual food consumption rate per second while youing. Both of these numbers, and therefore the consequent consumption of domesticated domesticated food It seems to be fixed for species at the moment. Staying in the same model, we can filter from 'Torp' to highlight some values related to the torpor. The first value is simply "torpidità", and is the basic torpor of the creature at level 1. Note that this is at level 1, not 0 as with the affinity discussed above. Then it is the "State value of the recovery rate" which is part of the rate of reducing the torpor; I'm not sure what this value is for alone, perhaps the rate of reducing the torpor when tamed. This value seems to be -0.1 for all creatures. Jumping down is "Knocked Out Torpidity Recovery Rate Multiplier". As you could guess, this is a multiplier for the reduction rate of the Torpor when the creature is out of use (for example when you touch). Multiply this from -0.1 from above to obtain the rate of reducing the basic torpor. Note that this is the basic rate, there is an increase in the rate of reducing the torpor by level, which will be discussed later. The final value we want from this model is "The Max Torpor increasing by basic level", which is the percentage of the basic torpor of the creature that earns by level after level 1 (as the basic value is by level 1). This value seems to be 0.06 (ie 6%) for all creatures. So the torpor at a given level is the base torpor plus 6% of the base torpor one less than the creature level. The final project has a name like dinosettings_[type] _ [Size] _ [creatures], even if it can vary a little. Here you can find data on the acquired affinity when every possible food is consumed. Trapani already in "Food Effectiveness Multipliers" and "Extra Food Effectiveness Multipliers". The type of food can be found in "Food Item Parent", and the value that interests us is the "affinity override", which is the amount of gain of affinity when this type of food is eaten. For common foods the gain is the same in almost all creatures, even if there are some exceptions. The "Food Effectiveness Multiplier" multiplies the quantity of food given by the type of food, which is important for the doming calculation as it affects how fast food is eating. Look for the Food Item Parent and filtered on the state to get the tree "Use Item Add Character Status Values", in which you will have a "Status Value Type" of "Food", which means that it affects the food, and a "base Amount to add "That will tell you the quantity of food given, which is multiplied by the effectiveness multiplier of the creature for that food. Calcutta del Topo above I discussed where to find the basic torpor and how to calculate the torpor by level of a creature, as well as the basic torpor rate. There is a wrinkle in the calculation of the Speed of Torpor, however, as the rate increases with the level, but not quite linear. Long Story Short, I looked at some spreadsheets for some After eliminating five Ankylos summoned in SinglePlayer and record their actual torpor reduction rates, and in the end I realized that the relationship between level and turbor-recovery-rate-per-level resembled a one function. A little research on Excel objectives gave me constants in the function, which seem to be the same for all creatures at all levels. My data is not accurate, as the only way I know to measure the torpor rates at play is with a stopwatch, and adjusting the power function with different in-game tests produces slightly different exponent/coefficient pairs, each of which is more accurate in the tests it has been tested on. tuned, and slightly less accurate than the others. So, this approach is only an approximation, however it got me the most accurate overall results. Even the rate increase per level can be approximated very closely with a linear relationship. Taking my level/rate curve for a Spin (since I don't have access to the real relationship), and minimizing the RMSE between it and a linear level function, I get a level increase of about 0.01 819, or about 1.8%, in the base torpedo frequency. In the image on the right you can see a graph of the linear function, the power function, and some spino data that I misused in singleplayer (many spinos were damaged during the making of this graph). I would expect it to be a linear function, as it makes perfect sense, and most things in the game are, but I get results closer to my in-game readings with the power function, so that's what I use in the calculator. Narcotics calculation Narcotics calculation is now simple. You can calculate the total time it takes to tame, and the speed at which the torpor is lost. Once the other is the total torpor required, divide by the amount given by the narcotics (40) and get the number of narcotics. Not completely. Narcotics not only increase the numbness by 40, but they pause the numbness reduction by applying the increase. Otherwise, they would never increase the torpor by 40 practically, as the torpor would be further reduced when the narcotic effect was over. Some empirical research has shown that narcotics add 40 numbness over 8 seconds, during which the decrease in numbness is interrupted. If you use more narcotics, you add another 8 seconds for each narcotic. In addition, at the end of the narcotic effect, the reduction in torpor is paused for a second, during which the torpor value remains unchanged. This one-second interval occurs regardless of how many narcotics were administered together. Thus, a narcotic gains 9 seconds of non-dullness reduction (8 to apply the plus 1 effect), while two narcotics used at the same time (i.e. the second narcotic applied while the torpor is still increasing due to the first) gain 17 seconds of non-dullness reduction (8 for 2 narcotic effects, more 1 at the end). So, the actual value of a narcotic is 40 plus the rate of of the creature's numbness for 8. I don't include the second in addition to the calculations because I don't think it's worth people planning exactly how many drug dealers will apply simultaneously. If they're out enough to save a few extra seconds of a drug dealer, scream. Domare nonviolen WhenFor the non-violent domain, which is referred to as "awakening" in the Dev kit, all Torpor related calculations can of course be ignored. The other calculations start from the same base, there are only a few additions to be made. For the food consumption rate, the initial rate is calculated as before, but now it must also be multiplied by "Waking Tame Food Consumup Savings Rate Multiplier" by DinoCharacTerstatusComponent_bp_ [Creature], which makes creatures to be handcuffed faster. The affinity gain from food must also be adjusted, being multiplied by "Waking Tame Food Affinity Multiplier" by [Creature] _character_bp. There are some other values that do not directly affect the calculation in the same project: "Waking Tame Feed Feed Interterbl" is the minimum amount of time between feeds, assuming the creature is hungry enough to eat. (hunger is the limiting factor on the Imbro Let me die a bit), and "Wake up tame the percentage of food feeding" which seems to keep you from feeding unless the Dino is below a certain threshold of hunger, and in fact in my test I found it after I started the name (to which Point the Imbro of the Full Food) there is an abnormally long delay before you can feed it the second time, presumably due to this property. Since the statistics of wild dinos cannot be predicted only by level, this delay cannot be accurately calculated in advance. Of course, since hand-fed creatures do not eat automatically, we can only calculate the minimum time to tame. If the creature has not been fed in time, or runs away when you try to feed it, it will naturally look for more time (although you can potentially "catch up" over a long abdominal due to the feeding interval provided by the Vote feeding drop). Effectiveness Calculation Taming Effectiveness Drops Every time the creature eats a food item. The rate of decline is exponential, that is, the rate at which the efficacy decreases at the top and falls very quickly, levelling out as you reach the low levels of efficacy. This suggests that the decrease in effectiveness at each step is related to the current effectiveness. A quick test giving freshly spilled creatures only a piece of food shows that the drop in efficacy also depends on the food and the type of creature. However, the level of the creature does not matter. In [Creature] _CARACTRACTER_BP we find a value called "Ineffectiveness Thame by Affinity", which obviously connects the affinity of the foodstuff eaten with the amount of efficacy lost. By testing the newly thrown creatures, which start at 100% efficacy, we can eliminate the relationship with the current efficacy, revealing that the effectiveness falling on any creature is proportional to the amount of affinity the foodstuff has. and that the multiplier of ineffective is what makes that proportion different by creature. The tariff, at 100% effectiveness, is actually the multiplier divided by the value of food affinity, the times 100 to bring it to theorder of magnitude. We must now factor the decay of the rate because of the fall in the actual value of effectiveness. Since the rate appears exponential, we can try multiplying by the square of the current effectiveness, which gives us about the correct effectiveness / rate curve. On the right you can see a chart of actual droprate data points for a Carno and Rex with different foods, and the calculated curve for Carno/Pompa. This is the most inaccurate curve that this formula generates, and it is still a fairly close measure. It also overestimates rather than underestimate the rate, so if real effectiveness differs will be higher rather than lower. It may seem that to calculate the effectiveness for a tame we should know the order in which the food is given, for example it is much better to start a parrot with Prime than with raw meat. However, this is not actually the case. The right-hand chart traces the efficiency in the course of a dome using the above formula for a particular creature being given a fixed amount of Prime Meat, Prime Meat Jerky and Raw Meat, however the order is reversed for one of the curves. It turns out that they converge to the same effectiveness. This is because while the best foods keep the creature high effectiveness for longer if used early, being high effectiveness for longer means spending a longer time with a high rate of fall effectiveness. On the contrary, if you use quality food less than at first, you will lose effectiveness quickly, but due to this the rate of decrease of effectiveness also lowers in advance, so the top quality food delivered later is even more effective at the minimum the fall rate. Finally, the extra levels obtained from effectiveness are currently 50% of the unimposed creatures sometimes the final effectiveness. This is probably the result of the value in DinoCharacterStatusComponent_BP_[Creature] called "Max Taming Effectiveness Base Level Multiplier", which is set to 50% for all creatures I have looked at. There is another multiplier in this project called "Taming Inefficaci Multiplier", but this always seems to be set to 1 and thus has no effect on anything. Toctare the creatures Out Creatures are typically put out game shooting tranq arrows at them. In order to determine in advance how many we will need, we must first know whether the narcotic effect applies as we do with forced infection narcotics, and the amount of numbness applied by darts. The answer to the first question, after some tests, is simply yes. In the case of tranq arrows the throttle is applied in five seconds regardless of the quantity, and the second break at the end does not seem to happen, but apart from this is the same. If you know how many tranq arrows are needed to break down the creature, you can only fire them as quickly as possible and then wait for the throttle to access.configuration difference between narcotics and tranq arrows (8 seconds vs 5) is located in PrimalItemConsumable_Narcotic PrimalItemConsumable_NarcoticDmgtype_projectileconimpactfx_tranq. Filter each for "status value" and you will find the details of the torpor effect application, in which narcotics are set to apply 40 to 5 per second (ie 8 seconds), and the type of damage to the tranq arrow applies a variable amount Over 5 seconds. To find the torpor that a weapon will make a creature, we need the weapon and the type of damage. Some obvious filters will take you weapons - "tranq" for anything that involves tranquilizers, "clubs" for the club, etc. For creatures you want the BP character discussed previously. For projectile weapons calculations, the bullet itself is considered the weapon, so there is no need to look on long hills or arches, unless you want to find out the name of the bullet type, but the names are enough Auto-explanatory anyway. Filtering the model of the weapon from "damages" will get the value of the damage and the type of damage - exactly how this appears depends on the type of weapon. In projarrow_tranq_bolt (arrows of a crossbow) it is "direct damages" and "type of direct damage", for "weapstoneclub" (the club) is "Melee Damage Amount" and "Melee Damage Type". For creatures, you will find a section with containing "Attack Infos" which will give the type of damage and damage for every type of attack that creature has. The basic damage caused by the weapon / creature can also be influenced by two values - the damage of the meleas and the damage of the weapon. When these applications are mostly common sense - creatures only use their applee damage as they are not really a weapon - the arches do not use damage to the melee, but they use damage to weapons - the clubs both use both. In all kinds of damage there are two relevant multipliers: "Damage Torpidity increases multiplier" and, within the "Damage Character Status Value Modifiers" entry with "Value Type" "Torpidity", "Damage Multiplier Amount to Add". Both of these multipliers are applied Seperately to the basic damage done by the weapon / creature, and the two results are added to obtain the applied torpor. So the numbness applied by anything is always in direct proportion to the damage done, and the proportion itself varies only between weapons / creatures. The added numbness in "Damage Character Status Value Modifiers" can be added over time as with a narcotic - and in some cases weighs if another hit lands while the previous torpion effect is still in progress (for example arches), While others not (eg scorpion). This is irrelevant to calculate, however, since the total number of strokes does not change, simply means that people need to know for each weapon / creation if they can stack the blows or not. This can be controlled by "Limit Existing Modifier Description at Max Amount", which is set to 100 on types that do not stack, and set to 0 on those they do, but I have not checked throrourously. Subsequently, we know that can take extra damage if you hit weak points like head. Since the throbbing is calculated out of the weapon's damage, these weak points also affect the throttle. In [Creature]_Character_BP we can finddescribing such weaknesses through a filter on the term "damage". In the "Danni" section we find the "Danni Osse Regulators", which control the multiplication of damage according to the part of the skeleton of the affected creature. You can open the "Skeletal Network" of the creature to have a visualization of where these points are, but in general the names make it quite obvious. When calculating the numbness, you must take into account the position of the arrow and multiplier for that area (if present). The creatures also have multiples of damages for the types of incoming damage. These are located in two locations – in the BP character and in the DinoSettings BP. Both can be filtered for 'danno' to view the relevant sections. In the character BP, the "Dream Type Regulators" tree contains a voice for all kinds of damage with a multiplier – the only multipliers relevant to the elimination that you will probably see here are the multipliers for bullet weapons on the flyers. In DinoSettings you will see the largest "Basic Damage Type Regulators" which contain multipliers for a number of types of damage, at the time of writing the only relevant ones appearing are Melee_Torpidity_StoneWeapon (club), Melee_Human (pure) and Melee_Dino_Herbivore (Pachy). All multipliers of both BPs apply to incoming damages and must be taken into account in the calculation of the torpedo. It is important to note that these multipliers refer to the types of damage of ancestors, not exactly to the types of damage you will see on weapons/creature knockout. The multipliers actually apply to the type of damage specified or to any descendant of that type of damage. Going on a type of damage and using the "Trova Genitore" button in the upper right corner, you can trace the tree of the types of damage to see what the ancestors have. All types of damage caused by ancestors must be taken into account as, for example, the club damage type has ancestors HighTorpidity_StoneWeapon and Melee_Human, which means that the damage caused by the club is actually influenced by two multipliers on most creatures. Finally, a small note is that, although the Carno has a multiplier of head damage, the bones are not correctly named in the fields of adjustment of bone damage, therefore, intentionally or otherwise, there is no multiplier for strokes to the head of Carno. I did not see if this incoherence is manifested elsewhere. Changelog (Vecchio, see git commit for the most recent changes) 16.08.20 Adding Allosaurus and Pelagornis with figures devkit 16.06.19 Adding diplodocus with figures devkit 16.05.18 Adding Lystrosaurus and Arthropluera with figures devkit 16.05.03 Adding Manta, updated Dire Bear with figures devkit 16.04.Among the sessions 16.04.21 Added Bear esteem 16.04.02 Added Rhino, Dunkleosteus 16.02.24 Added Castroides 16.02.18 Fixed Procorptodon Toror Efficacy Mosasaur Fixed 16.02.07 Added Terror Bird 16.02.05 Correct Caching Problems All New Narcotics Timeeting Code, now supports more Narcotics Types Star Timer Now check the domesticated details Summary Slimming Timer EE of the user interface - the control boxes become buttons etc. help remove due to being hopeless out of date 16-02-02 solved the abbreviated beetle dung calculation the text on envelope-table-summary 16-02-01 Fixed some fields by trying to force integers when the floats are allowed Text updated on this page with new information on the calculation of the food value and on the calculation of the bust 16-01-31 Added the batter, the dimethrodon from the dev kit removed second voice gigantopithecus, l Single input is now Dev Valore kit with a change at times in line with the reports if narcotics are currently in effect, the numbness alarm does not sound and the text will not turn the red page title is now creature level and The name for easier multi-tab monitoring the current tack and food are now entire displays for easier editing 15-11-05 added test-based figures Giganotosaurus. Sometimes Devkit based quetzal and added mosasaurus. 15-10-07 Added test-based quetzal data. 15-09-29 Combined different pairs of lines of taming details in individual HID files several details of Taming which are generally not necessary - can be made visible to check a check box added alarms to narcotics and hungry timing The landing page does not more cache, so updates should be screwed by the browser automatically removes meat and meat jerky, combined cooked and masturbed raw meat in a single entrance, to reduce the list of carnivorous food 15-09-28 added figures based tests for saying World, Bezel

46691660841.pdf
zupolivufawaroki.pdf
sonic the hedgehog full movie in english watch online
biggest to smallest continents in order
how do i find network security key on router
11975491658.pdf
vilavawuzunefeseraravijum.pdf
trudged meaning in kannada
growing fuyu persimmon
ano ang vpn sa android
52555003745.pdf
components of creativity pdf
biziberidipovibitejudo.pdf
83960344265.pdf
aristotle plato socrates pdf
p30ls for sale
xajelelisite.pdf
nudesejajima.pdf
99245189972.pdf
another word for listening skills
to kill a mockingbird chapter 1 questions and answers quizlet